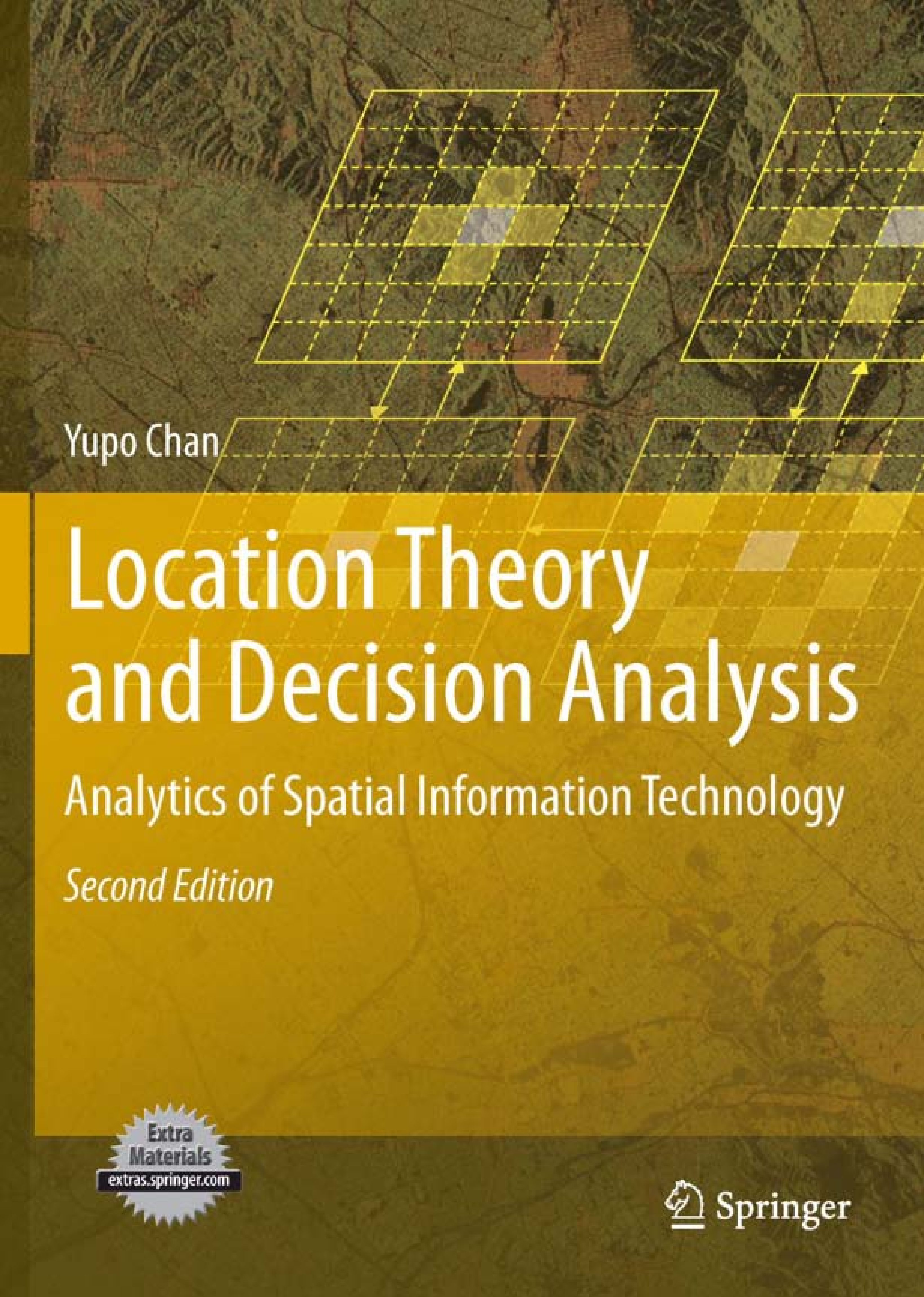


Yupo Chan

Location Theory and Decision Analysis

Analytics of Spatial Information Technology

Second Edition

Springer

Location Theory and Decision Analysis

Analytics of Spatial Information Technology

Second Edition

Yupo Chan
Professor & Founding Chair
Department of Systems Engineering
Donaghey College of Engineering and Information Technology
University of Arkansas at Little Rock

Springer

Prof. Dr. Yupo Chan
University of Arkansas at Little Rock
Department of Systems Engineering
Donaghey College of Info Sci. & Systems Eng.
South University Avenue 2801
72204-1099 Little Rock
Arkansas
USA
ychan@alum.MIT.edu

Additional material to this book can be downloaded from http://extras.springer.com.

ISBN 978-3-642-15662-5 e-ISBN 978-3-642-15663-2
DOI 10.1007/978-3-642-15663-2
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011921883

Cover design: deblik, Berlin

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Contents

SUPPLEMENTS ON THE CD/DVD

Self-Instructional Modules

Presentations (PowerPoint and PDF slides for instructors and students, organized by folder names)

Software and Data

Directory #1 BOOK: Software and data sets to support the analytics in the book

Directory #2 IMAGEFILES: Data files for image processing

Dedication

To my parents;
Susan;
Arthur and Mary

Preface to the Second Edition

Modeling and simulation are essential decision-making tools in many applications. Such decision making tools become increasingly important in today's increasingly service-oriented and ever changing economy. Recent events also suggest that many human decisions are transacted in a geo-spatial context. No where is it more obvious than the recent advances in telecommunication that bridge the geographic gap between anywhere in the world. This means that modern decision and risk analyses should take into full account the geo-spatial context in which they are being applied.

In today's information age, decision making also manifests itself in the prevalence of location-based services, as afforded by today's ubiquitous mobile or cell phone usage. As we commute to work, travel on business or pleasure, we make geo-spatial decisions, such as where to obtain gasoline, where to have lunch, and where to see the sights. It is also reflected in the popularity of such Internet engines as Google Earth. Google Earth combines satellite imagery, maps and the power of Google search to put the world's geographic and ancillary information at one's fingertips. In response, Microsoft and Yahoo are racing to transform online maps into full-blown browsers, organizing a diversity of information for better decision making.

Such information technology (IT) advances have given rise to the prevalence of E-commerce, and recently mobile commerce. When corporations device their business plans, they make geo-spatial decisions, such as where the prevalent suppliers and markets are. IT also found its way into E-government, where local, state and federal governments regulate and provide services within their geographic domain through the help of *informatics*, a broad academic field encompassing information science, IT, algorithms, and social science. To make better decisions, there is an increased emphasis on *analytics*, or the science of analysis. In other words, how an entity (i.e., business) arrives at an optimal or realistic decision based on existing data. One estimate suggests that 85 percent of existing data contain spatial attributes. "Mining" such spatial data is still an art, rather than a science.

Motivated by this trend, my personal conviction materialized in four books on this subject in the last decade. Obviously, what I wrote (and edited) were merely the first attempts to address a newly emerging and changing field. Concentrating on the current book, I am not sure my writing responded to the genuine needs of the readership. At the same time, the readership—by the nature of this field—consists of an *increasingly* diverse audience of many disciplines. For example, among the audience are practicing professionals and university students, constituting two of many disparate groups requiring different writing styles. Unlike a classic first book in calculus or economics, it is often a challenge to select, within a limited number of pages, the most cogent topics for decision-making in a geo-spatial context. Aside from exercising my best judgment, all I can promise is to improve my writing over time. This is the motivation behind writing this Second Edition on the modeling and simulation tools to support decisions based on geo-spatial data.

MOTIVATION FOR A SECOND EDITION

It has almost been a decade since the first edition of *Location Theory and Decision Analysis* appeared. During these years, I have received numerous feedbacks from both my students as well as from professionals. While the feedbacks have been positive, it is clear—for a number of reasons—that the book needs updating.

It goes without saying that any inadvertent mistakes that I know of in the First Edition have now been corrected in the Second Edition. The reader can agree that many of these mistakes are typographical errors that typically slip into the first printing of a book. While not fatal, they are certainly annoying, and it is best to get rid of them when they are found.

In 2005, a sequel to *Location Theory and Decision Analysis* was published by Springer, entitled *Location, Transport and Land-Use: Modelling Spatial-Temporal Information*. This companion book has been long waited by many readers, inasmuch as the two books are intimately related. With this sequel publication, it should become obvious to the readership that my plan has always been to provide the basic building blocks in *Location Theory and Decision Analysis*, serving as a survey of the decision analytic tools required for further study of the general subject of modeling spatial-temporal information. In other words, *Location Theory and Decision Analysis* is meant to pave the way for more specialized books on the subject, including *Location, Transport and Land-Use*. This is particularly important for an audience that comes from increasingly diverse disciplines (as mentioned)—engineering, management science, regional science, economics, geography, policy sciences, applied mathematics—just to name a few. Each discipline has been educated formally in a different way. For example, engineers and mathematicians are better prepared quantitatively, but sometimes lack the broader perspective that economists and management scientists have. The breadth of coverage in *Location Theory and Decision Analysis*, while ambitious, is simply to introduce concepts and techniques that may not be familiar to a particular discipline, in order for them to fully participate in the truly interdisciplinary field of spatial decision-making.

At the same time, there is a common body of knowledge all our readership relies on. My modest writing project is to outline this body of knowledge, which includes analytical techniques for modeling and simulation, as well as such IT as the computer software that support decision making. To underscore this point, I have added a subtitle to the Second Edition, *Analytics of Spatial Information Technology*. The complete book title now reads: *Location Theory and Decision Analysis*: *Analytics of Spatial Information Technology*—Second Edition.

Oftentimes, each discipline uses different jargons to connote the same idea. I have always wished to array these jargons and put them on a common denominator. This will facilitate dialogue between traditional disciplines. It is my opinion that this is a mandatory step for a more advanced study on a common subject. Whether done well or not, I am fully convinced that *Location Theory and*

Decision Analysis has taken the first step in this direction. It presents the basic building blocks for more in-depth studies in modeling and simulation, facility-location studies, transportation modeling, urban/regional planning, policy analysis, urban and transport geography, and other related disciplines.

What I tried and plan to do is really a "tall order." Fortunately, I was not alone to "deliver" this work. Based on the insightful feedbacks of my students, friends, and colleagues, here are the resulting changes I have incorporated in the Second Edition of *Location Theory and Decision Analysis.*

PEDAGOGY

The First Edition was written for a technical audience, by which I mean a number of *quantitative* or *analytical* disciplines—whether they be management scientists, engineers, social scientists, or applied mathematicians. I made a bold assumption on the background of the readership when I wrote the First Edition in year 2000. In hindsight, I might have assumed too much in the background of the audience. Not to be defensive, I was trying to introduce to other disciplines quite a few ideas and techniques indigenous to a specific discipline. In the lengthy appendices, I tried to review some basic mathematical techniques. All these have to be done in a limited number of pages, lest I would be reproducing separate books on decision making, simulation, optimization, microeconomics, geographic information system, satellite imagery, and the like.

As it turned out, while the First Edition might be good for professional use, my impression is that it was not totally responsive to classroom use. Meanwhile, IT has progressed leaps and bounds in the interim decade. Correspondingly, a main thrust of the Second Edition is to provide additional educational materials, including introductory Exercises and other pedagogic materials, to tailor toward educational programs that offer related courses on this subject. A major effort is also to strengthen the IT to support decision making in a spatial context. Toward this end, I have added two more chapters just on the topic:

Chapter 7—Analytics and Spatial Information Technology: Retrospect and Prospects

Chapter 8—A Software Survey of Analytics and Spatial Information Technology

In the First Edition, most of the exercises were either geared toward an advanced audience, or were open-ended case studies. Stressing the interrelationship among various topics in spatial decision-making, these "Synthesis Exercises" were organized into a single "appendix" at the end of the book, rather than segregated by chapters. In the Second Edition, I wish to supplement this approach with more introductory problems that are suitable for those who are exposed to the subject for the first time. These exercises are now prepared individually for each book chapter, as with most textbooks. I designed these new exercises as preludes and introduction to the more indepth "Synthesis" exercises.

In composing these new exercises, extra care has also been taken in leading the student step-by-step toward answering the questions posed. Two strategies were followed. First, a set of self-instructional "modules" was developed. Benefitting from my colleagues at Stony Brook University (State University of New York at Stony Brook), the fundamental principles of analysis were introduced in these modules. Trained in formal pedagogy, these colleagues—including Robert Seidman and Thomas Liao—initiated a set of courseware consisting of "fill in the blanks" modules. As first drafted by Gary Hom, these modules cover "Empirical Modeling," "Probability," "Probability Distribution and Queuing," "Graph Optimization," "Risk Assessment" and "Linear Programming." As I deployed and streamlined these modules over the decades, student feedbacks have been excellent from a diverse audience. Included in the CD/DVD and complete with solutions, this is one of the major milestones in this Second Edition. It is totally consistent with current educational emphasis on "interactive learning," in which students learn by engaging themselves, rather than passively absorbing materials from the instructor's lectures.

As I approach my forty-year milestone in higher education, I become convinced that students learn best when they are motivated. I can even go further by suggesting that motivation is the best teacher. It can overcome other impediments to learning, including incomplete background knowledge. The motivated student would simply take the time to catch up on her background, while a less motivated student would be condemned by such a impediment. In fact, the Self-Instructional Modules described in the last paragraph, aside from filling in some background gaps, are structured to put students from different background in the proper "frame of mind." My thesis is that learning the fundamentals of modeling and simulation requires the proper attitude, as learners of other disciplines do. If spatial decision making means an understanding of how humans function in today's location-based-services environment, the five modules are intended to cultivate this attitude. These modules examine not only the mathematical/analytical media, but also the socioeconomic genre that underpins the environ.

While the main body of the text concentrates on location theory and decision analysis, there are some computational aspects of model solution that readers may wish to review. The four technical appendices from the First Edition—System Stability, Statistical Tools, Markovian Processes, and Optimization Schemes—have been updated. While we assume a background of college algebra and calculus in our presentations, the Self-Instructional Modules serve as "icebreakers" to "ease" the less prepared readers into these rather condensed methodological reviews. As with other book appendices, these reviews are geared toward those specifically interested in the subject. To the extent that we are trying to cater to a multitude of disciplines, there will be a tendency from time to time, albeit infrequent in nature, to re-state the obvious. This is unavoidable in any attempt to reach a multidisciplinary audience. Each appendix is designed to be self-contained. References to outside sources and Chan (2005) are intended for further reading on the subject.

The second strategy I adopted in writing the Second Edition is actually an extension of the Self-Instructional Modules, by way of end-of-the-chapter *guided* Exercises. Instead of "filling in the blanks," the guided Exercises consist of more than just posing questions. They involve extended remarks on key points mentioned in the text. It is hoped that these pedagogic suggestions, integrated into the posed questions, are helpful and useful to the students. Also, they are intended to assist instructors in understanding why the particular exercise contributes toward learning the subject at hand. In summary, we fully subscribe to the philosophy of `active' or participatory learning and instruction. More will be said about this philosophy in the following paragraphs.

STATE-OF-THE-ART

Logically, another thrust of the Second Edition is the inclusion of new references published since 2001, when the First Edition was prepared. Best of all, these new references are fully integrated into the text for a state-of-the-art examination of the subject at hand. The sequel book *Location*, *Transport and Land-Us*e, published in 2005, supplements with additional references in the order of several orders of magnitude. The new references for the Second Edition are much more focused on geo-spatial decision-making technology, and are concentrated in the two new chapters, bearing the titles that speak for themselves: "Analytics and Spatial Information Technology" and the accompanying "A Software Survey." While the original edition has five software that comes with it, the software has now been expanded to include image-classification models in the PATTERN folder.

It is convenient that the two books—the current volume and the 2005 monograph—follow identical mathematical notations and terminologies, eliminating the task of getting used to different definitions and notations. To make it more convenient for the readers, I have reproduced the mathematical symbols at the back of this Second Edition, together with an updated glossary of Technical Concepts. I am particularly fond of the glossary, since it highlights the many fundamental concepts upon which the field is developed. The readers will recall that the glossary also provides a unifying delineation of cross-disciplinary terms in common use. It should supplement the book index in helping readers with unfamiliar terms.

Many have already discovered that there is a web site that supports the two books. Simply use an Internet search engine or e-mail me at my lifetime address ychan@alum.MIT.edu to locate the website. In the current Second Edition, I have printed solutions to the Self-Instructional Modules and selected solutions to the Synthesis Exercises. As a complement, the web-page posts a much larger collection of Synthesis Exercises. Of particular interest is an *Instructor's Guide* that provides solutions to one of every two of these exercises on the average. Moreover, it also has a digital version of the list of mathematical symbols and the glossary of Technical Concepts. It serves as a single reference site for readers of my two books.

SCIENTIFIC COMPUTING

While their outlines are printed in the Second Edition, I have decided to provide a bulk of the materials on the CD/DVD that accompanies the book. This will facilitate updating on a more frequent basis than the significant task of producing yet another printed edition of the book. As mentioned, a complete treatise on the computational IT support on the subject is documented in two new chapters. Chapter 7, entitled "Analytics and Spatial Information Technology," provides the parameters and the trend governing the computational support for spatial decision making. In so doing, it also summarizes the state-of-the-art in Location Theory and Decision Analysis. Chapter 8, entitled "A Software Survey of Analytics and Spatial Information Technology," gives a screened list of supporting software. All these are complemented with my expanded supporting software for the two books, which is also posted on my website.

In today's IT, the line between business software (such as spreadsheet and word precessing) and scientific computing (such as algorithms) has been blurred. Spreadsheets have been playing a prominent role in modeling and simulation, providing computations formerly reserved only for scientific computing. This is further complemented by Visual Basic for Applications, which allows decision support systems to be developed from spreadsheets, when assisted by a bit of programming. Meanwhile, software such as Scientific WorkPlace has put word processing and computer algebra system under one roof, complete with symbolic processing (as made available by Maple or MuPAD). MATLAB, a corner stone of scientific computing, has found its way into business applications, including built-in links for MS Excel database, financial applications, and a report generator. This blurring effect has been well recognized throughout the modeling community.

In this regard, I wish to clearly enunciate a philosophy I have been following. I like to minimize the reader's requirement to purchase expensive software. Most of the computing requirements for my exercises—both the new ones and the old ones—can be satisfied with the software that comes with the book CD/DVD, a regular Microsoft office suite such as Excel, or freeware such as OCTAVE, a public domain version of MATLAB. I also alert the reader, where appropriate, the availability of quality public-domain software. Chapters 7 and 8 alert readers to quite a few "freeware," mainly developed by educational and scientific institutions. Of particular interest is open-source software, which allows the user to build upon the available source codes. In fact, the software that comes with this book is open-source, in that I have included the sources codes, where applicable.

Where there is a suggestion for a commercial software, the purchasing decision is totally at the discretion of the reader. Today, many of the software vendors allow downloading of trial versions, facilitating the readers to judiciously select the ones that best suit their individual tastes. The bottom line is that the readers are not mandated to buy extra commercial software to use my book or to perform the exercises included in this book.

I have given some serious thoughts about packaging commercial software as part of the book. Being a new, interdisciplinary field, spatial decision-making has a broad and emerging scope of coverage. For that reason, the number of related software packages is diverse and numerous. As a result, a large number of them are needed in combination. Being a developmental field, most of the spatial IT software resides on university campuses and research institutions, and is in the public domain. There is also a thriving open-source community that share developmental software. Instead of packaging a suite of commercial software, I am providing the salient features of these modeling and simulation tools in Chapter 7, features that users should look for in a software. Accompanying this is the listing of relevant commercial and public-domain software in Chapter 8. The listing will be updated as new information technologies become available.

INFORMATION TECHNOLOGY

In the years since the First Edition appeared, IT has advanced by leaps and bounce. Many printed materials have been replaced by electronic soft files, using, for example, the Acrobat Portable Data Format (PDF). The Internet has also made it possible for students to rely on postings on the Internet, instead of going through printed volumes on library shelves. While there is still a definite place for printed words, serious thoughts need to be given to the best medium to document information. I believe that while easily dated materials should take advantage of the convenience of today's IT, information that is likely to survive the rigorous test of time still has its place in printed words.

Sometimes, there is a fuzzy area. I judge the lengthy Self-Instructional Modules are likely to stand the test of time, but their bulk speaks against placing them in the printed volume. They are best provided in soft copies on the CD, freeing up printed pages for more critical improvements in the text. Besides, their availability in a digital form will allow readers to port the information around different platforms. As a common denominator, however, all the archival documents on the CD/DVD are provided in PDF. Aside from its universality, the basic information can be extracted as text files for a relevant application of interest to the reader. The data sets, for example, can be directly extracted for the relevant computing platforms

As a companion of the current Second Edition volume are a set of PowerPoint presentations that cover the entire volume. These are included on the CD/DVD that accompanies the book. These presentations are intended for both the instructors and the students. While the presentations are based on the Figures and Tables in the text, they are supplemented by my teaching experience using the book as a text. Some of the presentations are animated, as enabled by the current features of MS PowerPoint. To cut across the media, the presentations are also prepared in PDF files in case the readers choose to go beyond MS products. Going beyond the presentations, available to the readers are the instructional materials for multiple courses that have adopted the current volume as a text, ranging from Decision and Risk Analysis to Optimization. For the latest set of course notes, simply drop me a line at ychan@alum.MIT.edu

In general, extensive use of soft instead of hard copies has a final, important advantage. It saves production cost by cutting down the number of printed pages. This ultimately reduces the sales price and saves money for the readers.

USE OF THIS BOOK

In my forty years of teaching, I have taught in diverse instructional programs, including systems engineering, operations research, civil engineering, policy sciences, plus technology and society. Since this book reflects my experience, I believe it is suitable for use in each of the aforementioned programs. Universal to all Systems Engineering programs is a course in Decision and Risk Analysis. Toward this, the current volume is as good a textbook as others, with the pervasive geo-spatial information required in today's work place. Many civil engineering programs require students to take a course in systems engineering. I believe this book, with its focus on spatial modeling and simulation, is ideally suited for that purpose, given civil engineers are responsible for constructed infrastructures that dot the landscapes of modern society. Management Science/Operations Research programs often offer a course on Decision Analysis, for which the book can serve as a text. Industrial Engineering programs offer a course on Facility Location, for which this text may be suitable. Known by many names, policy sciences and management refer to treatment of public/societal issues through analytical means. This discipline is often housed within management or public administration schools. Irrespective of where it resides, this book can serve that audience in foundation courses in modeling, particularly when it has geographic implications.

Here are some example courses that maybe served by the contents of the current volume. For this purpose, let us excerpt from various university catalogues:

□ "Decision and Risk Analysis" (graduate course in Systems Engineering Certificate and Master's Degree, Stevens Institute of Technology)

Cover analytic techniques for rational decision-making that addresses uncertainty, conflicting objectives, and differing risk attitudes. Learn about modeling uncertainty, rational decision-making principles, representing decision problems with value trees, decision trees and influence diagrams, solving value hierarchies, defining and calculating the value of information, incorporating risk attitudes into the analysis, and conducting sensitivity analyses.

□ "Engineering System Design" (undergraduate course in Civil & Environmental Engineering Department, MIT)

This class provides an introduction to quantitative models and qualitative frameworks for studying complex engineering systems. Also taught is the art of abstracting a complex system into a model for purposes of analysis and design while dealing with complexity, emergent behavior, stochasticity, non-linearities and the requirements of many stakeholders with divergent objectives.

□ "Facilities Location, Layout and Material Handling" (undergraduate course in Industrial Engineering Department, Texas A&M University)

Analytical treatment of facilities location, physical layout, material flow and handling, combined with heuristics algorithms to assist in the design of production/service facilities; fundamental concepts applied through a sequence of design projects.

□ "Methods of Policy Analysis" (master's course in the Heinz School of Public Policy & Management, Carnegie-Mellon University)

This course is designed to teach students practical techniques for analyzing public policy problems and developing effective solutions to them. Students will learn a series of interrelated methods of policy analysis and gain experience in analyzing realistic policy cases using those methods. Students are assumed to have mastered these skills: Applied Economic Analysis, Empirical Methods, Management Science, Policy & Politics, Organizational Design and Implementation, Financial Analysis, Management Information Systems, Professional Writing, and Professional Speaking.

ACKNOWLEDGEMENTS

A large number of my friends and colleagues helped in my production of the First Edition. In compiling this Second Edition, I am also indebted at least to the following individuals worldwide who have stepped forward since my publication in 2001. I am indeed grateful for their guidance, advice, and/or encouragement:

Peter Astor, USA

Sérgio Barreto, Portugal

Mike Cain, United Kingdom

James P. DeAngelis, USA

Satinderpaul Singh Devgan, USA

Marek J. Druzdzel, USA

Karunarathna Panagamuwa Gamage, Germany

Dennis Hanseman, USA

Samuelson W. Hong, Taiwan

Emily Johns, USA

Martin Landré, Netherlands

Stefan Nickel, Germany

Amelia Regan, USA

Ahmad Siamil, USA

Minghe Sun, USA

Laith Zumot, Jordan

Antanas Zilinskas, United Kingdom

Meanwhile, all remaining errors or omissions are my personal oversight. If you have any suggestions to offer, I would like to hear from you at ychan@alum.MIT.edu. Thanks in advance for your courtesy.

YUPO CHAN

Preface

Location Theory and Decision Analysis is tailored toward upperclass and graduate-level courses that include location decision making. It includes the fundamental theories and analysis procedures of that process. With these fundamentals carefully and comprehensively compiled, it is amply suited for courses such as management science, operations research, economics, civil and environmental engineering, industrial engineering, geography, urban and regional planning and policy sciences. The book also serves as an overview of the relationship between location, transport, and land use decisions. As such it introduces more advanced topics as documented in Chan (2005) and Easa and Chan (2000).

This book is unique in that it integrates existing practical and theoretical works on facility location and land use. Instead of dealing with individual facility location or the resulting land use pattern alone, it provides the underlying principles that are behind both types of models. Of particular interest is the emphasis on counter-intuitive decisions, which are often overlooked unless deliberate steps of analysis are taken. Being oriented toward the fundamental principles of infrastructure management, the book transcends the traditional engineering and planning disciplines, where the main concerns are often exclusively physical design, fiscal, socioeconomic, or political considerations.

Employing contemporary quantitative models and case studies, the book discusses the siting of such facilities as transportation terminals, warehouses, nuclear power plants, military bases, landfills, emergency shelters, state parks, and industrial plants. The book also demonstrates the use of satellite imagery, computer-based data-retrieval technologies (such as geographic information systems), and statistical tools for forecasting and analyzing implications of land use decisions. The idea is that land use shown on a map is necessarily a consequence of individual, and often conflicting, siting decisions.

The analytical community has made significant progress in recent years in the basic building blocks of spatial analysis. Current models have captured accurately many of the bases of facility-siting decision making—proximity to demand, competition among existing facilities, and the availability of utilities and other institutional supports. Throughout this text, accessibility (as afforded by transportation) and infrastructure support (as provided by utilities and sewers) are used as determinants of location decisions. Competitive and statistical determinants that are not based on accessibility alone are also covered.

However, a novel feature of *Location Theory and Decision Analysis* is the recognition that in today's service economy, the traditional concepts of accessibility need to be broadly interpreted. Evidence indicates, for example, that half of the shopping currently done is by mail, telephone, or the Internet. Thus the definition of "a trip to the shopping mall," and hence the conventional judgment in siting a retail facility, need to accommodate such a change. "Global reach" redefines the concept of accessibility and distance in all sectors of the economy, including E-commerce, international corporations, and even the defense community. Half the globe away now means a few hours of flight time or seconds of telecommunication time. Conversely, congested streets can make cross-town travel almost impossible, and thus encourage telecommuting. Again a redefinition of accessibility and hence the conventional wisdom in office site selection is required. The theme of change carries throughout the book, serving to unify many of the spatial location models discussed.

The advances in remote sensing imagery and geographic information systems today facilitate much of spatial analysis. Electronic devices, such as satellites, sensors, computers, and telecommunications technology, make the collection and processing of data much faster, which in turn assists in the problem solving process. The book discusses how information can be stored in such a way that it can be directly translated to a format for real-time decision making. This means simple and transparent models that are database compatible and require minimal data manipulation in the solution process. These models become the tools for analysis and decision making. *Location Theory and Decision Analysis* gives the reader a comprehensive insight into the use of these tools—identifying, assembling and utilizing the important information for problem solving, rather than prescribing verbatim software instructions.

ORGANIZATION OF THE BOOK

As mentioned, this book contains a comprehensive review of the fundamental principles. Questions such as why facilities locate where they do and why population and employment activities distribute on the map as they do are answered. The first few chapters include the underlying determinants of facility location and land use, as well as the techniques that are essential to analyze these location decisions. In addition, these chapters discuss databases from remote sensing and geographic information systems (GIS), statistical tools for data analysis and forecasting, optimization procedures for choosing the desirable course of action, and multicriteria decision-making techniques to tie the entire analysis procedure together. Key concepts in economics, one of the most important disciplines in explaining the organization in space, are also reviewed.

The first five chapters—which include economics, descriptive and prescriptive techniques, and multicriteria decision making—constitute an excellent quantitatively oriented survey course in this field. If needed, the appendices provide for a review of the mathematical tools. Where there is room in the curriculum, a more advanced treatment will include the "Remote Sensing and GIS" chapter. While the first five chapters redefine location by such concepts as telecommuting, Chapter 6 drives it home. In this last chapter, new ways to store, organize, process, and transmit spatial data are reviewed.

Location Theory and Decision Analysis purposefully accommodates the different technical backgrounds and career objectives of its readers. For example, spatial economics principles are introduced in Chapter 2, allowing the non-economists to acquire the basic economic concepts that underlie much of the location literature. It serves as an excellent overview of the entire book. As another example, multicriteria decision making is reviewed in Chapter 5, with an emphasis on how it assists in location decisions. It includes discussion of state-of-the-art concepts and technology that may not be familiar to those outside the fields of management science and operations research. For example, I illustrate how an obnoxious facility, such as a noisy airport, can be located by taking into consideration all the stakeholders concerns. Most importantly, liberal numerical examples and graphics are used to get the point across. My diverse background, which spans technical consulting firms, government, academia, and the defense community, enables me to communicate with different audiences in terms of a common language. Beyond the classroom, professionals who seek an update on the fundamentals on location decisions will find this book helpful. The professional audience will find the crosscutting discussion of technical concepts in Appendix 5 particularly helpful, since it unifies the findings from different disciplines.

Exercises and case studies are used throughout the book. Rather than a set of mechanical calculations, the exercises and case studies are designed to extend many of the concepts covered in the book. They also play an important role in integrating the many diverse principles advanced in the text. One objective of the exercises is to challenge the readers creatively to use the data sets and computer software that come with *Location Theory and Decision Analysis*. While the basic exercises are well structured, readers are often asked to perform their own case studies, using the data sets if desired, and arrive at open-ended results. For the sake of synergy, all the exercises are placed together at the end of the book, rather than included separately at the end of individual chapters. To assist both instructors and students, answers to the exercises are available on my web site. Please contact me by email at ychan@alum.MIT.edu for information about the web site. Students and professionals should enter in the Subject line: REQUEST FOR SAMPLE SOLUTIONS, and instructors should enter: REQUEST FOR INSTRUCTOR'S GUIDE.

SOFTWARE

A CD-ROM provided with the text provides sample software. The main purpose of the CD is to supplement the basic ideas covered in the text. Aside from extensive databases, it contains software to implement some of the basic concepts presented. It also challenges the reader to investigate further through hands-on experiences with case studies. In view of the rapid progress in information technologies and to avoid obsolescence, the book is not specifically tied to a single generation of information technology. Rather, the book is problem-oriented and provides a set of procedures and a set of data for analysis that can transcend the technological evolution. Hands-on experiences are discussed with respect to the basic models employed, rather than the particular software or hardware.

One software program used for processing remote sensing images (courtesy of Dr. T. S. Kelso) illustrates some of the spatial statistical concepts and GIS. The remainder are software implementations of some of the facility-location and land-use concepts discussed in this book. While the book introduces the various analytical techniques in a pedagogic fashion, the software provides practical implementations. The programs are therefore not purely for the classroom; they have real potential for everyday, operational use.

1. All files on the CD are ASCII-text files. Where possible, both source codes and executable codes are given—mainly for the ease of execution and modification by the users. Program documentation is included as README files.
2. While references are made to supporting software for extended use of some of the programs, all programs are self-contained, and they have been developed or refined by the author and his associates. The programs do not require supporting software or language compilers.

As mentioned, sample data sets are provided to allow demonstration of the software. Most of the data are drawn from real-world case studies.

The programs have been extensively tested, but still there can be no absolute guarantee of faultlessness. It is impossible for me to provide any programming support for the software, but I am keenly interested in and would appreciate any feedback from users regarding their experiences with the programs or the book. To provide your comments, simply contact me by email at ychan@alum.MIT.edu and include in the subject line: SUGGESTIONS FOR THE BOOK.

ACKNOWLEDGEMENTS

Location Theory and Decision Analysis is a result of over two decades of work that benefited most importantly from my students but also from academic and professional colleagues around the world. The group includes reviewers of the actual manuscript.

Suraj M. Alexander, University of Louisville

Mark S. Daskin, Northwestern University

Dinesh S. Dave, Appalachian State University

Walter Fabricky, Virginia Polytechnic Institute and State University

James Farris, Golden West College

Paul Gray, Clarmont Graduate University

Ronald Klimberg, Boston University

Duane Marble, Ohio State University

Pitu B. Mirchandani, University of Arizona

Norbert Oppenheim, CUNY—City College

Stephen Robinson, University of Wisconsin

William B. Widhelm, University of Maryland

Others who aided my efforts were collaborators in various studies, such as a paper that won the 1991 Koopman Prize Award of the Institute for

Operations Research and Management Science (INFORMS). In addition to explicit references in the text, I wish to acknowledge their input. The following group by no means constitutes everyone who helped to make this book possible. However, all these people rendered invaluable guidance to this writing project.

Steven Baker, Air Force Academy; David Boyce, University of Illinois at Chicago; T. Owen Carroll, SUNY at Stony Brook; Emilio Casetti, Ohio State University; F. Stuart Chapin, University of North Carolina at Chapel Hill; Jarad L. Cohon, Carnegie Mellon University; Noel Cressie, Iowa State University; Richard deNeufville, Massachusetts Institute of Technology; John W. Dickey, Virginia Polytechnic Institute and State University; O. Day Ding, California Polytechnic State University; Kenneth J. Dueker, Portland State University; Alan G. Feldt, University of Michigan; Richard L. Francis, University of Florida; Jon Fricker, Purdue University; Richard Gagnon, North Shore Community College; William Garrison, University of California-Berkeley; Bruce L. Golden, University of Maryland; S. Louis Hakimi, University of California-Davis; Elise Miller-Hooks, Pennsylvania State University; Chin S. Hsu, Washington State University; Zhimin Huang, Adelphi University; Arthur P. Hurter, Northwestern University; John J. Jarvis, Georgia Institute of Technology; Edward J. Kaiser, University of North Carolina at Chapel Hill; Ralph Keeney, University of Southern California; Tschangho J. Kim, University of Illinois; Thomas S. Kelso, Air Force Institute of Technology; David. H. Marks, Massachusetts Institute of Technology; Edward K. Morlok, University of Pennsylvania; James G. Morris, University of Wisconsin-Madison; Srinivas Peeta, Purdue University; G. L. Peterson, Northwestern University; Peter Purdue; Naval Postgraduate School; Stehpen Putman, University of Pennsylvania; Essam Radwan, University of Central Florida; Charles Revelle, Johns Hopkins University; Brian D. Ripley, University of Washington; Morton H. Schneider, Johns Hopkins University; Thomas Sexton, SUNY at Stony Brook; Ralph E. Steuer, University of Georgia; Eric Vanmarcke, Princeton University; Steven C. Wheelwright, Harvard University; John A. White, Georgia Institute of Technology; Jeff R. Wright, Purdue University; Ping Yi, University of Akron

G. Leonardi, Instituto di Analisi dei Sistemi ed Informatica, Rome, Italy; Peter Nijkamp, Free University, Netherlands; S. Occelli, Instituto Ricerche Economico Sociali, Turin, Italy; Atsuyuki Okabe, University of Tokyo, Japan; M. B. Priestley, University of Manchester Institute of Science and Technology, United Kingdom; Peter M. Pruzan, Copenhagen Business School, Denmark; C. S. Bertuglia, Instituto Ricerche Economico Sociali, Turin, Italy; Barry Boots, Wilfrid Laurier University, Canada; Erhan Erkut, University of Alberta, Canada; Jakob Krarup, University of Copenhagen, Denmark; William H-K Lam, Hong Kong Polytechnic University, Hong Kong, China; Gilbert Laporte, University of Montreal, Canada; G. A. Rabino, Instituto Ricerche Economico Sociali, Turin, Italy; Jacque Thisse, Ecole Nationale des Ponts et Chaussees, France; Roger W. Vickerman, University of Kent at Canterbury, United Kingdom; George O. Wesolowsky, McMaster University, Canada; Alan G. Wilson, University of Leeds, United Kingdom; Maurice Yeates, Queen's University, Canada

Wayne Allison, Warren, Rhode Island; Kurt Ardaman, Fishback, Dominick, Bennett, Steper & Ardaman, Orlando, Florida; Frank Campanile, Dayton, Ohio;

Raul Meda Dooley, City Bell, Argentina; Carolyn B. Doty, Automated Sciences Group, Oak Ridge, Tennessee; A. Ruth Fitzgerald, Fitzgerald and Halliday, Hartford, Connecticut; Terry Cronin, Intelligence & Electronic Warfare Directorate, Warrenton, Virginia; Paul Guyer, Guyer Santin Inc., Sacramento, California; Walter G. Hansen, Frederick R. Harris Inc, Washington D. C.; Harvey Haack, Pennsylvania Department of Transportation; Rubin Johnson, OR Concepts Applied, Whittier, California; Frank Kudrna, Jr., Kudrna and Associates, Chicago, Illinois; Douglass B. Lee, Volpe Center, U. S. Department of Transportation; Ira S. Lowry, Housing and development consultant, Pacific Palisades, California; Robert E. Machol, Washington D. C.; Reddy Mannur, U. S. Department of Justice, Washington D. C.; David Merrill, Air Mobility Command, Scott AFB, Illinois; Larry Mugler, Denver Regional Council of Governments, Denver, Colorado; Phil D. Patterson, Department of Energy, Washington D. C.; Armando Perez, Camp Dresser & McKee Inc, Miami, Florida; Alan B. Pritsker, Pritsker and Associates, West Lafayette, Indiana; Joe H. Mize, Stillwater, Oklahoma; M. Rahman, JHK & Associates, Seattle, Washington; Earl Ruiter, Cambridge Systematics, Cambridge, Massachusetts; Howard Slavin, Caliper Corporation, Boston, Massachusetts; Bruce Spear, Volpe Center, U. S. Department of Transportation; Curtis Travis, Oak Ridge National Laboratory

I also benefited from my association with the Massachusetts Institute of Technology, Kates, Peat, Marwick & Co., Pennsylvania State University at University Park, U.S. Congressional Office of Technology Assessment, State University of New York at Stony Brook, Washington State University, and the Air Force Institute of Technology. Don Hoover is thanked for his role in editing and compiling the various iterations of this manuscript. Bradley Smith helped me with debugging the software packaged on the CD. I wish to express my gratitude to all those at South-Western College Publishing/Thomson Learning™, including Senior Acquisitions Editor Charles McCormick, Senior Development Editor Alice Denny, Senior Marketing Manager Joseph Sabatino, and Production Editor Anne Chimenti.

The strength of this work is one of synthesis — cutting across disciplines, backgrounds and experiences — precisely where this field is heading. The diverse backgrounds of these friends and colleagues in academia, government, and private industry made my job as synthesizer that much more streamlined. These people greatly assisted me in achieving the goal of informing the reader of the knowledge that falls within a specific discipline but also across other disciplines — in as readable, yet as precise and practical, a form as possible. Their assistance did not end with this volume. Chan (2005) includes these advanced topics: facility location; measuring spatial separation; simultaneous location-and-routing models; generation, competition and distribution in location-allocation; activity allocation and derivation; chaos, catastrophe, bifurcation and disaggregation; spatial equilibrium and disequilibrium; spatial econometric models; spatial time series; and spatial-temporal information. Easa and Chan (2000) includes these additional topics: trends in spatial databases, spatial decision-support systems, GIS integration with analytical models, and incorporating real-time information. It is through these additional discussions that one can fully realize the power of synergism.

YUPO CHAN

ABOUT THE COVER

This radar image shows the massive urbanization of Los Angeles, California. The complete image extends from the Santa Monica Bay at the left to the San Gabriel Mountains at the right. Downtown Los Angeles is on the right side of the textbook's cover. The complex freeway system is visible as dark lines throughout the image. Some city areas, such as Santa Monica in the upper left, appear red due to the alignment of streets and buildings to the incoming radar beam.

The image was acquired by the Spaceborne Imaging Radar-C/X-band Synthetic Aperture Radar (SIR-C/X-SAR) onboard the space shuttle Endeavour on October 3, 1994. SIR-C/X-SAR, a joint mission of the German, Italian and U. S. space agencies, is part of NASA's Mission to Planet Earth. The radar images illuminate earth with microwaves allowing detailed observations at any time, regardless of weather or sunlight conditions. The multi-frequency data will be used by the international scientific community to better understand the global environment and how it is changing. The SIR-C/X-SAR data, complemented by aircraft and ground studies, will give scientists clearer insights into those environmental changes that are caused by nature and those changes that are induced by human activity.

REFERENCES

Chan, Y. (2005). *Location, transport and land-use: Modelling spatial-temporal information.* Berlin and New York: Springer.

Easa, S.; Chan, Y., eds. (2000). *GIS: Applications in urban planning and development.* Reston, Virginia: ASCE Press.

1

Introduction

"Where the telescope ends, the microscope begins. Which of the two has the grander view?"
Victor Hugo

I. OBJECTIVES

This book has three basic objectives. The first objective is *to identify the observed regularities in location decisions.* This involves examining and answering questions such as: Why do public and private facilities locate themselves the way they do? What factors do real estate developers consider when picking sites for development? Why do people live in a certain location, and why do they often work in a location different from where they live? Why are focal points such as airports, terminals, and depots situated at certain nodes in a network? Throughout this book, we will try to answer some of these questions, so that readers can judiciously locate facilities and guide development toward desired goals.

While we often take notice as to why certain facilities are placed in certain areas, we get as many explanations about such location decisions as the number of experts we ask. Each seems to offer a plausible explanation. Such explanations can be any combination of economic, technical, social, political, and behavioral reasons, not to mention such philosophies as **feng shui**—which roughly translated means "location and orientation [of a facility] with respect to the elements of nature" (Love, Morris, and Wesolowsky 1988). Are there really discernible patterns about these location decisions? Many of us have observed that ports and cities of the world are often located on major trade routes, usually at the confluence of rivers, a convenient deep sea harbor, or where railroads come together. Scientists envision future habitats in the galaxies being located at **Lagrangian points**—locations that are stable enough that space stations located there, when perturbed by slight impacts, will restore their position after reasonable oscillations. Based on these examples, it stands to reason that there may be some location patterns one can discern. These patterns, when observed to be consistently recurring in one area after another, are referred to as **regularities**. These regularities are not anywhere as precise as scientific laws, nor can they often be explained in terms of cause-effect relationships. One event does not necessarily occur because of a previous event. As a result, we have to go by the observed **patterns**

Y. Chan, *Location Theory and Decision Analysis*, 2nd ed., DOI 10.1007/978-3-642-15663-2_1,

only and to treat those recurring patterns merely as some generally agreed upon facts. From there, analytical models can be built to reflect these premises. The first objective of this document then, is to understand, in a systematic manner, the regularity with which different location decisions are made, so that systematic procedures can be defined to anticipate similar situations that may arise in the future.

We should quickly point out there is a difference between the systematic analysis proposed here and **comprehensive,** or **holistic planning,** which goes under different names such as **morphology, concurrency,** or **planning theory**. That body of knowledge, while extremely valuable, has been treated in excellent texts elsewhere, including those that are required reading in such professional examinations as those of the American Institute of Certified Planners (AICP), and in such documents as *Land Use* by Davis (1976). This book aims at a different area that is by definition more narrowly focused. We ask more specific questions, such as "how does transportation affect location decisions?"; "how does infrastructural support influence development of a certain area?"; or "how does transportation combined with infrastructural support affect facility location and land use?" In other words, we examine one factor at a time, one criterion at a time, and the cumulative effects rather than the simultaneous effects of all factors across all criteria. Distinction is also made between the treatment here and an approach taken by two notable publications—one by the American Society of Civil Engineers (1986), the *Urban Planning Guide,* and another by Brewer and Alter (1988), *The Complete Manual of Land Planning and Development.* The *Guide* is an excellent document that discusses a whole host of planning topics, ranging from waste to energy planning, with a design flavor as an undertone. Brewer and Alter's publication is a comprehensive description of site layout planning and design. As illustrated by examples at the end of this chapter, the focus here is on analytics, with quantitative model building as a key instrument. Thus this book serves as a useful companion to such documents as the *Guide* and represents an area that has not been covered sufficiently for many who feel the need for state-of-the-art analytic tools to make capital-intensive location decisions—the step prior to detailed design.

The second objective is *to review the operational analysis techniques that have been applied in the field*. In this regard, we report on case studies that span a number of user groups—from public and private facility location to land development. For example, we would look at the factors that go into the location of a nuclear power plant in seismically active areas in California, the location of state parks in the greater New York metropolitan area, the choice of distribution centers for military logistics, the siting of satellite tracking stations in Canada, target location in search and rescue missions, and the land development in several major North American cities, including a systematic study of bifurcation development in a medium-sized city: York, Pennsylvania. We also examine case studies around the world, including the economic impact of the Kansai International Airport outside Osaka, Japan. The common theme is how location regularities and spatial impacts can be quantified in a set of procedures or models.

The third objective—*to be able to stand back and critique some of these modeling experiences*—requires asking whether they have been successful and valid. In other words, what are the assets and liabilities of the various techniques that have been employed? Perhaps one can think of this book as a consumer's guide to location analysis and land use models. A user can look up the price tag of using a particular model, and also the benefits, specifically regarding the problem being solved. The only time that a model or analysis procedure can help is when the user is fully aware not only of its strong points, but its shortcomings as well. Only under those circumstances can an engineer, an analyst, a planner, or a manager employ the most appropriate tool toward the problem at hand, and avoid overkill with exotic technology, below-par performances with an outdated tool, and misfits between problem and analysis tools in general.

What are the more visible results and benefits from reading this text? For engineers, analysts, planners, and managers, the question is easy to answer. As long as infrastructure represents a major capital expenditure and supports economic well being and quality of life, this book serves the important role of articulating investment in these infrastructure improvements. Such infrastructure may include tracking stations, depots, terminals, roads, factories, warehouses, hazardous facilities, office buildings, and housing. Both in public decisions and in corporate planning, the analytical skills discussed in this book can mean savings or benefits in terms of a huge number of dollars. To students and researchers, this book serves as a useful compendium of spatial analysis techniques. It is a comprehensive collection, and the presentation style is pedagogic, starting from the basic building blocks to the more advanced concepts. We point out the commonalities among models used to locate facilities one at a time and to forecast the development pattern in an entire area. In this regard, it is a unified volume on **spatial science**—defined here to mean the analytical techniques that explicitly recognize the spatial elements in a study. The term spatial science, when used in this context, encompasses the traditional disciplines of facility location, transportation, logistics, land use, regional science, quantitative geography, and spatial economics. This book introduces to students and specialists in each of these disciplines the broader perspective as viewed from collective wisdom—a perspective that is absolutely essential to furthering the art of spatial science.

II. DETERMINANTS OF LOCATION

One goal of this book is to uncover the observed regularities of location decisions, in other words, the apparent underlying forces that shape development. We shall examine four major determinants of location.

A. Technological Factors

The first determinant refers to physical principles that govern location and infrastructural supports such as highways, airports, railroads, power supply, sewers, and irrigation. These supports make the functioning of the facility possible. Notice that these go beyond the availability of transportation and utilities. The example about building a space station drives this home. Only Lagrangian points in spatial mechanics will allow the location of a permanent habitat/resupply station in deep space at which spaceships can dock conveniently with the assurance that it is a stable station that can survive the impact of objects. Likewise, satellite tracking stations must be where visibility is at its best to observe the desired orbits most of the year. It stands to reason that a station too far north in the Northern Hemisphere will be unsuitable to track satellite orbits around the equator, not to mention that infrastructural support such as roadways and utilities will be scanty in these arctic regions. When the American West was developed, the railroad was the key instrument. Today, in the Midwest of the United States, one can still trace the location of towns in regular intervals along the rail lines on the prairie. They were apparently developed from water refilling stations required for the steam locomotives of the day. The separation represents the length during which all the water carried on a train evaporated—a technological factor in its truest sense.

B. Economic and Geographic Factors

A person lives at a location convenient for carrying out daily activities, both work and non-work, commensurate with the ability and willingness to pay for the corresponding residential cost. For those who cannot afford the prime locations, housing a little bit further away is the only choice. A host of theories exists to explain this phenomenon, including land rent and location theories. On a historical basis, cities have located on trade routes, perhaps due to accessibility to markets. To command a competitive edge in today's retail market, warehouses are often situated in the midst of the demand, where consumers have easy access to stored goods through the retail outlets. The most graphic example may be in emergency planning. Quick, efficient medical evacuation of the wounded dictates a judicious placement of hubs through which the injured can be quickly transported and eventually delivered to hospitals for medical care.

C. Political Factors

Zoning represents an institutionalized consensus in the community regarding the legitimate use of the land. Fiscal and jurisdictional considerations are also quite common. During the latter part of the 20th century, there have been free enterprise zones designated by the People's Republic of China to manufacture and conduct business with the free world. Some of these are located across the border from Hong Kong and Macao. These zones enjoy special jurisdictional and fiscal privileges—incentives for investment and workers. Finally, there are eminent political decisions for location as well. For example, the Dallas-Fort Worth Airport in the United States sprawls across two counties, apparently for political reasons—which in part explains its having a huge horizontal layout rather than a more vertically integrated structure. On a larger scale, many guidelines are enacted as legislation. The location of airports, for example, is subject to numerous environmental regulations. Brewer and Alter (1988) and Chapin and Kaiser (1979), among others, have a good review of the national, state, and local legislation that governs land use in general.

D. Social Factors

Dominance, gradient, and segregation, centralization and decentralization, and invasion and succession are social factors that determine location. Humans tend to congregate into communities. On the other hand, they tend to segregate themselves for certain other reasons, which results in the reservation of certain land accessible only by selected groups. Thus there are segregated regions reserved for colonial citizens in a newly discovered land to the exclusion of natives of the land. Certain public facilities are segregated between women and men for privacy reasons. Between the phenomenon of togetherness and separation, all the shades exist in between. This explains to some extent the myriad of development patterns that we see through recorded history. These social and behavioral factors vary depending on the values of the time and the context of the culture. They are somewhat difficult to quantify in a set of systematic procedures.

III. THE ROLE OF ANALYSIS

Some explanations of the perplexing issues raised can be found by the judicious employment of analysis techniques. Obviously, analysis of the problems posed above requires a set of very specialized skills. The techniques required of the analyst include **descriptive** and **prescriptive tools**. Descriptive tools are the techniques that echo location regularities that we observe around us. They are the representation of observed patterns by way of such methodologies as simulation and statistics, or more causal explanations such as regional economics. Through the use of computers, one can build a mathematical replica of the scenario and use it to test out alternative policies—much like architects will build a scale model of a building for study in a studio. Graphic display of information, afforded by today's geographic information systems, greatly facilitates such analysis (Thrall, McClanahan, and Elshaw-Thrall 1995; Transportation Research Board 2000; Ozbay and Mukherjee 2000).

Prescriptive tools, on the other hand, try to identify a course of action for decision makers. For example, to achieve the community goals and objectives, one specifies a set of policies and plans by means of goal-directed methodologies. A mathematical model can be formulated, from which one obtains a blueprint for future development. As with descriptive models, computers are often utilized to operationalize optimization models of various sorts, including those that take into account multiple criteria, echoing a pluralistic decision-making environment typical of location decisions. Advances in computational techniques have made it practical to identify desired courses of action or facility locations, which was impossible only ten years ago. While part of the advances have been due to the computational machinery, our understanding of prescriptive techniques has also made dramatic gains in the past decades.

Analysis can reveal counter-intuitive results that can easily be overlooked if such a set of rigorous thoughts are not carried out. This pertains obviously to complex situations where there are just too many factors to consider for the unaided mind to comprehend. What is more interesting is that they may arise in rather simple situations as well. We will demonstrate a couple of these below, which hopefully make a strong case for the analysis procedures advanced in this text.

A. Airport Example

Suppose a common airport is to be built to service New York City and New Haven, Connecticut—a distance of about 80 miles. Where is the best location considering the combined populations of the two cities—with approximately 14 million in New York and 2 million in New Haven? Notice the question asked here, being a narrowly focused one, is simply how to reduce the travel requirement for all the 16 million residents of the area—in terms of total **person-miles-of-travel (PMT)**. Most people who are asked the question responded by saying that the airport should be somewhere in between the two cities. Some even pointed out that it should be closer to New York than to New Haven, since New York is a larger city. The more technically minded calculated that it should be 10 miles outside Manhattan and 70 miles away from New Haven on the major highway that connects the two cities.

The correct answer in this case is that the airport, from a purely accessibility standpoint, should be as close to New York as possible. It is that location that will require the lowest PMT. To show this, just pick three possible locations:

- □ halfway between New York and New Haven, resulting in a travel requirement of (40 × 14 + 40 × 2) or 640 million PMT.
- □ 10 miles outside New York and 70 miles from New Haven, resulting in a PMT of (10 × 14 + 70 × 2) or 280 million.
- □ located right at New York and a full length of 80 miles from New Haven, resulting in (0 × 10 + 80 × 2) or 160 million PMT!

When presented with this result, people quickly pointed out that it is impossible to locate a new airport at New York, since there is simply no land. Others pointed out that environmentally speaking, no one will accept an airport close to New York City. But that was not the question. The question—which still appears in black and white above—simply focuses on one aspect: the total PMT!

We will come back to this in a case study later, where we will point out that those having knowledge of linear programming—a prescriptive technique—will readily recognize an extreme point—either New York or New Haven—as the site for the airport, not somewhere in between.[1] We will at that time bring in other considerations, including the environment, and show how the location may change as a result of these additional factors. In other words, we answer the question for the accessibility factor, then the environmental factor and so on—building up the complexity as we move along, rather than facing them simultaneously as in more holistic planning methodologies. In Chapter 5 under "Interactive Frank-Wolfe Example," we illustrate how decision-theoretic tools can assist, without an explicit knowledge on valuation, in considering noise impacts.

B. Manufacturing Plant Example

Another example equally illustrates the role of analysis as advocated here. Suppose a major manufacturer opens an additional plant in Home Town, with a payroll of 1000 workers. What will the future population and employment increase be in Home Town? We further know that each household in Home Town has 2.5 people on the average, of which there is only one breadwinner. For every five additional people, one more support service employee is required. In other words, there are multiplier effects on the economy, wherein one dollar of payroll generates more than its value in the local economy. The manufacturing employees will require support services such as shopping, medical, recreation, and so forth, involving new employees who also bring in their families who again require more services.

Table 1.1 ECONOMIC FORECAST OF HOME TOWN

Time increment	Basic employ	Basic-employ pop	Support-service emp	Support-service pop	Total employ	Total pop
1	1000	2500	500	1250	1500	3750
2	—	—	250	625	1750	4375
3	—	—	125	312.5	1875	4687.5
4	—	—	62.5	156.25	1937.5	4843.75
5	—	—	31.25	78.13	1968.75	4921.88
6	—	—	15.63	39.06	1984.38	4960.94
7	—	—	. . .	. . .	. . .	. . .

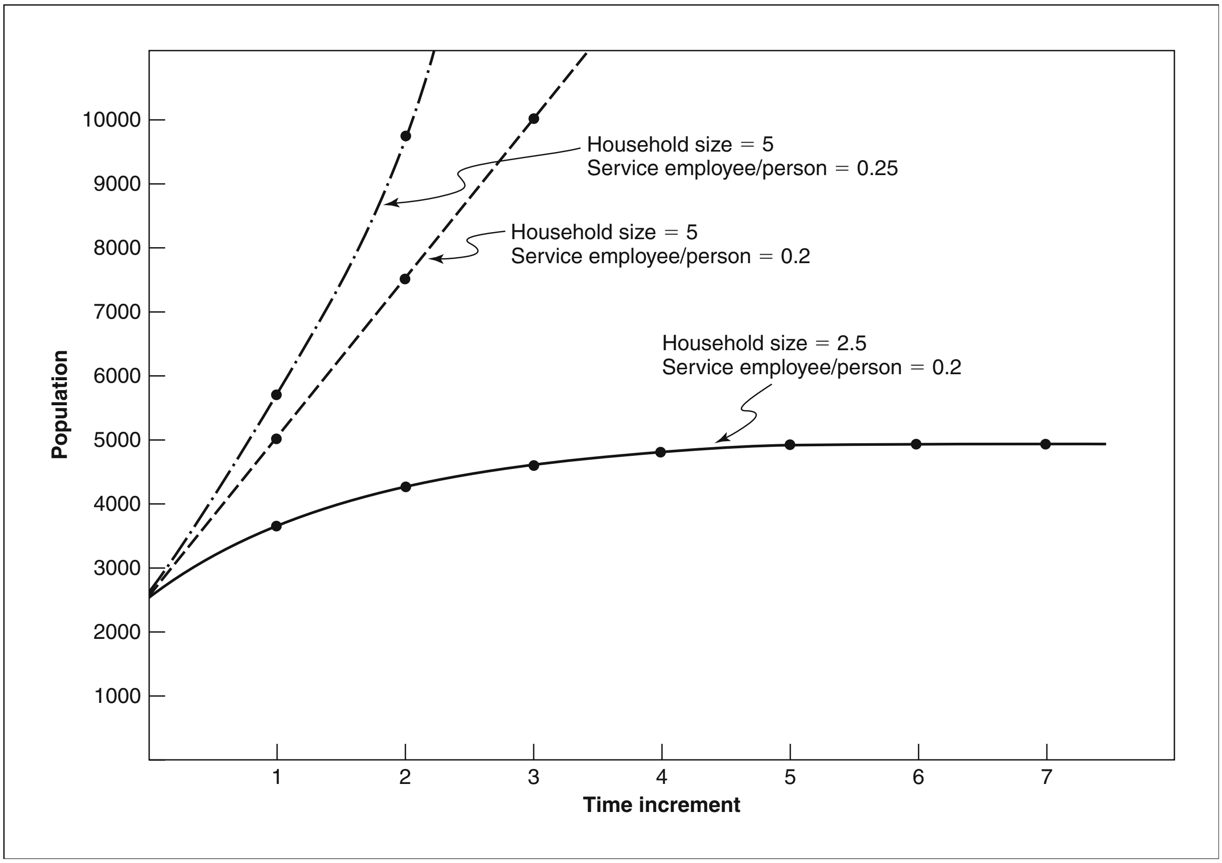
According to the parameters given above, a moment's reflection will show that the 1000 new manufacturing jobs will bring into town 2500 people, including dependents. These 2500 people will also require support services in Home Town—including medical, shopping, recreation, and so forth—and generate 500 secondary jobs. These secondary service jobs bring into town another 1250 population (500 × 2.5), including employees and family members. (In this case, every five people require one secondary service employees.) Now the total new employment in town is (1000 + 500) or 1500, and the total new population is 3750 (2500 + 1250). The process goes on as shown in Table 1.1, eventually stabilizing at about 2000 additional employees and 5000 additional people.

Figure 1.1 depicts the growth profile of Home Town in terms of population and employment. The growth profile stabilizes in time period 7. On the same figure is shown the growth profile when household size is increased from 2.5 to 5. In this case, the growth will perpetuate forever, as shown by the straight line of Figure 1.1. When the support service requirement is raised from 1 to 1.25 employees for every 5 people, totally uncontrolled growth will result, as shown again in Figure 1.1. Apparently, any slight increase beyond the watershed points of 5 people in a household and 1 service employee for every 5 people will fuel the fire of growth to a fury. On the other hand, family size a tiny bit smaller than 5 or service requirement less than 1 employee in 5 results in a stabilized growth in due course. The watershed point is an important piece of information for all who are interested in the future of Home Town. A technical term for the dividing line between growth versus stagnation is **bifurcation**. Without descriptive analysis such as the above, these bifurcation points are not obvious to simple, intuitive reasoning.

C. A Combined Example

Now, combining the above examples, if 1000 new jobs are added both to New Haven and to New York City, if the average family size is 2.5 people in New York and 5 people in New Haven, and if there is 1 service employee for every 5 people

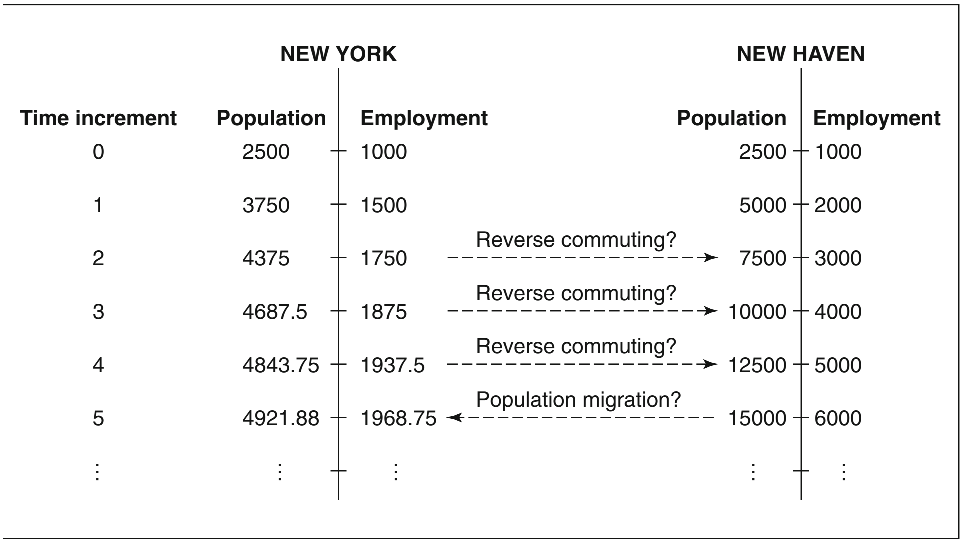
Figure 1.1 BIFURCATION IN POPULATION GROWTH



in both places, New Haven will experience unlimited growth while New York will be stagnating. It does not take long for the labor force of New York to see job opportunities in New Haven and respond to them in terms of reverse commuting. Nor does it take long for the unlimited growth in New Haven to outgrow its physical or infrastructural capacity. Given the growth in New Haven will have to go somewhere, this will possibly mean the spread of wealth back to New York. Figure 1.2 represents this interaction between the cities schematically. The time increment is on the vertical axis and the spatial interaction is on the horizontal axis. Different growth profiles, combined with the physical limitation to unlimited growth, result in an interesting development pattern between the two cities.

With a changing demographic profile, the location of a regional airport will have to be reconsidered. We have already demonstrated that from purely an accessibility standpoint, the airport should be located at the more populated of the two cities. Now with New Haven enjoying unlimited growth while New York is stabilizing at 14,005,000, it is only a matter of time before New Haven will surpass New York in terms of population (assuming the physical limitation to growth has yet to be reached). The regional airport will eventually be located at New Haven instead of New York. The interesting, somewhat counterintuitive, fact is that the best location for the airport will switch abruptly the minute New Haven has one person more than New York—no sooner and no

Figure 1.2 ECONOMIC INTERACTION BETWEEN NEW YORK AND NEW HAVEN OVER TIME



	NEW YORK			NEW HAVEN	
Time increment	Population	Employment		Population	Employment
0	2500	1000		2500	1000
1	3750	1500		5000	2000
2	4375	1750	Reverse commuting? →	7500	3000
3	4687.5	1875	Reverse commuting? →	10000	4000
4	4843.75	1937.5	Reverse commuting? →	12500	5000
5	4921.88	1968.75	← Population migration?	15000	6000
⋮	⋮	⋮		⋮	⋮

later. The moment that the New Haven population exceeds New York's is a bifurcation point, at which precipitous changes occur in the fundamental behavior of the system.

IV. ANALYTICAL TECHNIQUES

These examples drive home the point that analysis is an indispensable supplement to intuition in capital-intensive location decisions. These examples are merely abstractions of case studies that will be presented in detail in later chapters and in Chan (2005), where the highly simplified situations used above are protracted into the multicriteria and pluralistic decision-making process (Massam 1988) common in location debates. Suffice to say here that sophisticated analytical techniques have been developed in recent decades to perform these studies. These techniques are based—by and large—on operations research, statistics, economic analysis, and systems science. The contribution of this book is not just the collection of these techniques, but more importantly the extension of them into the spatial context. Thus the well-known extremal point optimality of linear programming (LP) is now extended from the Euclidean space of LP into the physical map of the Northeast, including the metropolis of New York and New Haven. It will be seen that when a triangle of three cities—say New York, New Haven, and Newark, New Jersey are involved, the complexity of locating a regional airport compounds many fold, resulting in the classic brain teaser: the Steiner-Weber problem. The airport can now be located—again based on proximity—at any of the three cities or in the interior point of the triangle, as will be discussed in Chapter 4. Anyone who has worked with this problem can testify to the fact that the Steiner-Weber problem is not simply an extension of LP—it goes well beyond.

The same can be argued about the Home Town development example. As seen above, the simple aspatial statement of the problem can quickly get complicated as we extend to two cities interacting with one another: say between New York and New Haven. It will be shown in sequel that the underlying theory is **input-output analysis**, a branch of knowledge economists since Leontief have developed to explain trade between such economic sectors as manufacturing, service, and housing. Including the spatial element into input-output analysis, however, compounds the model significantly, raising a whole host of conceptual and model calibration problems as evidenced in the well-publicized Lowry-Garin derivative models. When fully developed, several important factors have to be reckoned with here, including spatial competition such as in an oligopoly market consisting of several well-defined competitors, and hence supply-side investment strategies to stimulate subareal and areal economic growth. Intimately related is the fundamental assumption about factor substitution—for example, to what extent can labor be traded off against capital in the spatial production process. In other words, can labor savings be effected by better equipment and production facilities at certain sites? Simply put, the Lowry-Garin derivative models are more than just a straightforward extension of aspatial input-output analysis.

Given the complexity of including a spatial dimension, is there a fundamental basic building block of spatial interaction: the foundation that enables spatial generalization of most analysis techniques? Yes, there is, in fact, a simple spatial law, credited to Tobler (1965) which states that "Everything is related to everything else, but closer things are more related than distant things." The power lies in the beguiling simplicity of the statement, which finds its way into pervasive applications in facility location, land use, and image processing. It turns out that geographers, transportation planners, electro-optics researchers, and statisticians have all worried about this phenomenon for decades, if not centuries. At the core is the concept of a neighbor, which is intimately tied to the definition of proximity or **spatial separation** (Nicholas et al. 2004). Spatial separation in this case goes well beyond just the Euclidean metric or bee-line distance. It is best thought of as a price system that organizes location decisions, much as the familiar monetary price that allocates scarce resources in microeconomics; a higher price discourages consumption while a lower price stimulates consumption.

Alternatively, **proximity** is the metric that establishes correlation among entities in space, such as pixels in an image. Thus accessibility in urban commuting takes on a very different light than proximity between two pixels (picture elements) in a satellite image (Faghri et al. 2001). Yet in some ways, the fundamental principles governing both are remarkably similar in concept, namely Tobler's first law as outlined above. Furthermore, both the urban planner and the remote sensing analyst have the common goal of monitoring land use.

The **gravity model**, which relates spatial interaction as a direct function of activity levels and inverse function of spatial separation, is one of the popular implementations of Tobler's law. Thus more traffic is found between high-density residential and employment centers that are close together than in lower density ones that are further apart. Likewise, satellites that monitor pollution will observe pollutants dissipating in inverse-square relationship to the point source. Calibration of the gravity model, however, is by no means simple, often necessitating a fundamental re-examination of an entire array of basic statistical principles (Sen and Smith 1995). By now, the reader should have a taste of the complexity of spatial science.

The above represents merely a few examples of interest. In modeling and simulating spatial systems, these additional questions often arise (Larson and Odoni 1981):

1. Given the coordinates of an incident requiring dispatching of a service unit (such as an ambulance or a fire truck), in which geographical subdivision did this incident occur?
2. Which available service unit is the closest one (in terms of travel distance or travel time) to a request for assistance?
3. Which zones have areas in common with a particular police precinct?
4. Which city blocks lie closest to each new voting center, so that voters can be assigned to the closest voting precinct?
5. Which ZIP-code areas contain segments of a particular highway or railroad?
6. Given that a refuse incinerator is to be built and that it has a certain geometrically described pattern of smoke dispersal, which voting districts will be affected by the pollutants?
7. Which geographical subdivisions have overlapping parts with the various noise contours (reflecting different noise exposure) that will result from a proposed airport runway?

Notice that all these questions have spatial attributes. In the following chapters, we will take up these questions one at a time. In Chapter 6, we will talk about image processing and geographic information systems that constitute some methods to answer these questions, using pattern recognition and districting. While humans can look up the answers manually on a map, however, software has to be *instructed* to answer these questions. For computer-based modeling and simulation—a focus of this book—it invariably requires us to carry out these two fundamental analytical steps:

Step 1: To decide in which zone the event location will be, obtain a sample from the probability mass function depicting the relative likelihood of events among zones.

Step 2: To identify the exact location of the event, obtain a sample from a uniform distribution over the zone selected (assuming that an event is equally likely to occur anywhere within a zone).

Aside from the discussions forthcoming in subsequent chapters, Larson and Odoni (1981) amounts to a dearth of literature that codifies these steps. It provided a way to generate events that are geographically distributed among and within zones in accordance with some pre-specified probability law. They also addressed this reverse question: "Given a point with a coordinate (x, y), in which geographical zone is it contained?" A "point-polygon method" was introduced to answer this question. Let us focus on simple polygons, or polygons with no overlapping sides. Suppose such a polygon has n clockwise ordered vertices, a set of procedures can be devised based on a sorting algorithm with computational complexity $O\ (n \log_2 n)$.

Using these procedures, one can also check whether zones overlap with each other and, if so, to identify the pairs of zones that overlap, as well as whether the overlap is partial or if one zone is fully included in another. It is also possible to identify the polygon that forms the intersection of two overlapping zones. Unfortunately, there are no shortcuts for doing this faster than the straightforward method. In the worst case, each side of polygon A will intersect every side of B (and vice versa), there will be a total of n^2 intersections (if A and B are n-sided polygons), and therefore the computational effort must also be at least $O(n^2)$.

V. CONCLUDING REMARKS

While making facility location and land use decisions is truly an art, there appears to be an information base that can or should explain, perhaps one factor at a time, these decisions. Factors range from technological and political to economic and social. Our purpose in this book is to trace the effects of these factors, not necessarily in a holistic manner, but rather by trying to identify the consequence of each decision. Prescriptive and descriptive methodologies play a role in clarifying these decisions. For example, some of the phenomena are counter-intuitive, and an analytical framework will extend our intuition a long way toward seeing details that are not apparent to the unaided mind. The examples of airport location and the corridor development between New York and New Haven should drive this home.

Today's headlines are filled with competition for industries to locate in a certain locale, state, or nation, perhaps for both economic and political gains. In fact, facility location decisions have faced humankind throughout history. A familiar example can be found in the development of the steel industry in the United States. While iron ore was found in the convenient open pits of the Mesabi Range in Minnesota, coal was plentiful in Pennsylvania. Considering the amount of coal required, it constituted the more expensive of the two commodities to transport. Thus, we saw the historical development of steel mills in Pittsburgh, Pennsylvania; while iron ore was shipped through Duluth, Minnesota, coal was collected at Pittsburgh via the Monongahela River. Perhaps this is another example of the LP application in airport location, where the facility is located at either one of the extreme points, rather than somewhere in between.

In today's economy, where globalization and technological innovation become dominant factors, it is critical to ask how location conditions vis-a-vis structure and strategies of the firm play a part in the competitive market. When the innovation process is explained in terms of product cycle and diffusion, relevant location factors are stressed and a hierarchical pattern of innovation in space is arrived at. On the other hand, evolutionary and network theories point to the relevance of historically evolved firm structures and strategies. Empirical evidence seems to accommodate both schools of thought (Todtling 1992). In some firms, there appeared to be a pronounced differentiation of innovation across space, such as concentration of research and development and product innovation in the largest agglomerations. However, strong innovation activities, corresponding more with the evolutionary model, were in addition identified in old industrial areas and newly industrialized rural areas. It is required, more than ever, to discern the relevant factor that plays the pivoting role under these mixed development patterns, particularly when location decision becomes paramount.

Facility location and land use decisions are highly capital-intensive and highly valued. In an emergency situation, a location decision often makes the difference between security and danger. Nowhere is it more apparent than the ongoing debate on hazardous facility location, where a fine line exists between perception and reality. Recent advances in multicriteria decision-making techniques can shed some light in the debate between the proponents of and the opposition to such facilities. In short, location decisions have long-term effects on the regional and interregional economy and profound implications on the quality of life. Modern analysis techniques can shed light on the matter and can have significant rewards in more informed choices.

VI. EXERCISES

Self-Instructional Module: EMPIRICAL MODELING

(to be found on the attached CD/DVD)[2]

Models range from the simple (one equation) to the complex, requiring computer processing. (A model of the national economy of the U.S., for example, can easily consists of hundreds if not thousands of equations.) This module works with models of one equation, serving as an introduction to modeling, such as the urban growth and decline example in Chapter 1, and the econometric models covered in Chapters 2 and 3 of the textbook. After completing this module the reader should:

(a) Be exposed to model construction based on empirical data.
(b) Become familiar the use of power/exponential functions in econometric modeling—a useful background for spatial development modeling.
(c) Understand the somewhat counter-intuitive properties of power or exponential functions through log-linear transformation using semi-log graphs.

This Modeling module serves as an excellent introduction to the often counter-intuitive findings offered by mathematical models, such as the urban growth and decline and airport-location example in Chapter 1. The body of this text is replete with examples of these counter-intuitive findings.

It is the author's wish that many readers' curiosity would be aroused by these findings, particularly regarding the explanation of these unusual phenomena. As such, those who are mathematically inclined may wish to review the Appendix entitled "Control, Dynamics, and System Stability," which offers a number of mathematical explanation of unexpected system behaviors.

Problem 1: Further Discussions on Table 1.1

The calculations in Table 1.1 can be simplified if we employ some algebra.
Let us define these terms:
Basic employment = E^B = 1,000
No. of persons per household = f = 2.5

No. of retail/service employees generated per person = $a = 0.2$
Support service employment = E^R
Total employment = E
Total population = N

With these terms, Table 1.1 can be expressed mathematically as Table 3.1, which is reproduced here for convenience

Time	E^B	Basic emp pop	E^R	Service pop	E	N
1	E^B	fE^B	afE^B	afE^B	$E^B(1 + af)$	$fE^B(1 + af)$
2			$a^2f^2E^B$	$a^2f^3E^B$	$E^B(1 + af + a^2f^2)$	$fE^B(1 + af + a^2f^2)$
3			$a^3f^3E^B$	$a^3f^4E^B$	$E^B(1 + af + a^2f^2 + a^3f^3)$	$fE^B(1 + af + a^2f^2 + a^3f^3)$
•			•	•	•	•
•			•	•	•	•
m			$a^mf^mE^B$	$a^mf^{m+1}E^B$	$E^B(1 + af + a^2f^2 + a^3f^3 + \bullet\bullet\bullet)$	$fE^B(1 + af + a^2f^2 + a^3f^3 + \bullet\bullet\bullet)$

When $m \rightarrow \infty$, it forms a "geometric series," as will be explained in book Chapter 3, Section III:

Total employment is $E = E^B(1 - af)^{-1} = 1000\ (1 - (0.2)(2.5))^{-1} = 2000$

Total population is $N = fE^B(1 - af)^{-1} = 2.5 \times 1000\ (1 - (0.2)(2.5))^{-1} = 5000$

- **(a)** Reconstruct Table 1.1 for total employment and population when $a = 0.2$ and $f = 5$.
- **(b)** Reconstruct Table 1.1 for total employment and population when $a = 0.25$ and $f = 5$
- **(c)** Does the process stabilize in (a)? How about (b)?

Problem 2: Further Discussions on Airport Location

The book discussion on airport location was based on the "median" concept, or we wish to minimize the sum of person-miles of travel for both the New York and New Haven residents. There is another concept in facility location, namely that of the "center" concept, in which the facility is located where the distance for the person furthest away is minimized. In other words, we wish to soften the inconvenience to those living far away.

Let us define x as the distance the airport is from New York. Under the center concept, the weighted distance from New York is $14x$. On the other hand, the weighted distance from New Haven is $2(80 - x)$. A Table can be constructed for different values of x:

x	$14x$	$2(80 - x)$	Max	Min(Max)
10	140	140	140	140
20	280	120	280	280
30	420	100	420	420
40	560	80	560	560
50	700	60	700	700
60	840	40	840	840
70	980	20	980	980

It appears that the best airport location will now be 10 miles from New York, or 70 miles from New Haven. Just to be careful, we can check for $x = 5$, which yields $14x = 70$ and $2(80 - x) = 150$. This is no better than what we have currently. It appears that we have the "optimal" location as is.

(a) For Figure 1.2, please compute the center location of the airport at time increments 0, 1, 2, 3, 4 and 5.

(b) Discuss the stability of this airport location compared to that obtained by the "median" concept.

ENDNOTE

[1] An introduction to linear programming is contained in Chapter 4 and also in Appendix 4.

[2] The answer to this Modeling module is attached at the end of this textbook.

REFERENCES

American Society of Civil Engineers (1986). *Urban Planning Guide*. New York: ASCE.

Antoine, J.; Fischer, G.; Makowski, M. (1997). "Multiple criteria and land use analysis." *Applied Mathematics and Computation* 83:195–215.

Brewer, W. E.; Alter, C. P. (1988). *The complete manual of land planning and development*. Englewood Cliffs, N.J.: Prentice Hall.

Chan, Y. (2005). *Location, transport and land-use: Modelling spatial-temporal information*. Berlin and New York: Springer.

Chapin, F. S.; Kaiser, E. J. (1979). *Urban land use planning*, 3rd ed. Urbana, Ill.: University of Illinois Press.

Davis, K. P. (1976). *Land use*. New York: McGraw-Hill.

Faghri, A. J.; Lang, A. J.; Henck, H. E. (2001). "An AL-based hybrid system for locating of park-and-ride facilities." *Pre-Print CD*, Transportation Research Board 80th Annual Meeting, January, Washington, D.C.

Larson, R.; Odoni, A. (1981). *Urban operations research*. Prentice-Hall, Englewood Cliffs, New Jersey.

Love, R. F.; Morris, J. G.; Wesolowsky, G. O. (1988). *Facilities location: Models and methods*. New York: North-Holland.

Massam, B. H. (1988). "Multi-criteria decision making (MCDM) techniques in planning." Vol. 30, Part I of *Progress in planning*, edited by D. Diamond and J. B. McLoughlin. Oxford, England and New York: Pergamon Press.

Nicholas, M. A.; Handy, S. L.; Sperling, D. (2004). "Using geographic information systems to evaluate siting and networks of hydrogen stations." *Transportation Research Record*, No. 1880:126–134.

Ozbay, K.; Mukherjee, S. (2000). "Web-based geographic information system for advanced transportation management system." *Transportation Research Record*, No. 1719:200–208.

Sen, A. K.; Smith, T. E. (1995). *Gravity models of spatial interaction behavior*. Berlin and New York: Springer-Verlag.

Thrall, G. I.; McClanahan, M.; Elshaw-Thrall, S. (1995). "Ninety years of urban growth as described with GIS: A historic geography." *Geo Info Systems* (April):20–45.

Tobler, W. R. (1965). "Computation of the correspondence of geographical patterns." *Papers and Proceedings of the Regional Science Association* 15:131–139.

Todtling, F. (1992). "Technological change at the regional level: The role of location, firm structure, and strategy." *Environment and Planning A* 24:1565–1584.

Transportation Research Board (2000). *Using geographic information systems for welfare to work transportation planning and service delivery*: *A handbook*. Report 60, Transit Cooperative Research Program, National Academy Press, Washington, D.C.

Yun, D-S; Kelly, M. E. (1997). "Modeling the day-of-the-week shopping activity and travel patterns." *Socio-Economic Planning Sciences* 31, No. 4:307–309.

2

Economic Methods of Analysis

"Two and two the mathematician continues to make four, in spite of the whine of the amateur for three, or the cry of the critic for five."
James McNeill Whistler

Most of the underlying theories of facility location and land use models are basically economic concepts, and many of their input/output variables are economic measures. To understand these relationships better, a general knowledge of economic concepts and methodology is helpful. We recognize that theories have been offered by economists to explain the growth and distribution of industrial activities in an area. It is insightful to summarize their experiences—particularly the theories used in regional and interregional economics. This includes such concepts as economic-base theory (or export service theory) of gravitational interaction and theory of interregional flow. Through such a review, one sharpens the focus on the validity and limitations of these analysis methodologies.

We will also outline the basic techniques for evaluating the impact of a proposed policy on transportation systems, utility systems, and zoning codes. When an evaluation measure is often phrased in terms such as cost, benefit, equity, and efficiency, a clear understanding of these terms is necessary. Conversely, when indicators such as opportunity and quality of life are output from the model, they are much more meaningful if one can relate them to the economic theories of cost/benefit and equity/efficiency. Such an understanding would help the inquiring mind to understand the assumptions based upon which the measures are derived. Finally, for the model builder, the review of economic methods would help them configure better models and submodels.

I. ECONOMIC CONSTRUCTS FOR ACTIVITY ALLOCATION AND FORECASTING

Econometricians have been forecasting economic activities such as population and employment for a long time. Two types of forecasting methodologies can be broadly classified—forecasting on the basis of **cross-sectional data** versus that

Y. Chan, *Location Theory and Decision Analysis*, 2nd ed., DOI 10.1007/978-3-642-15663-2_2,

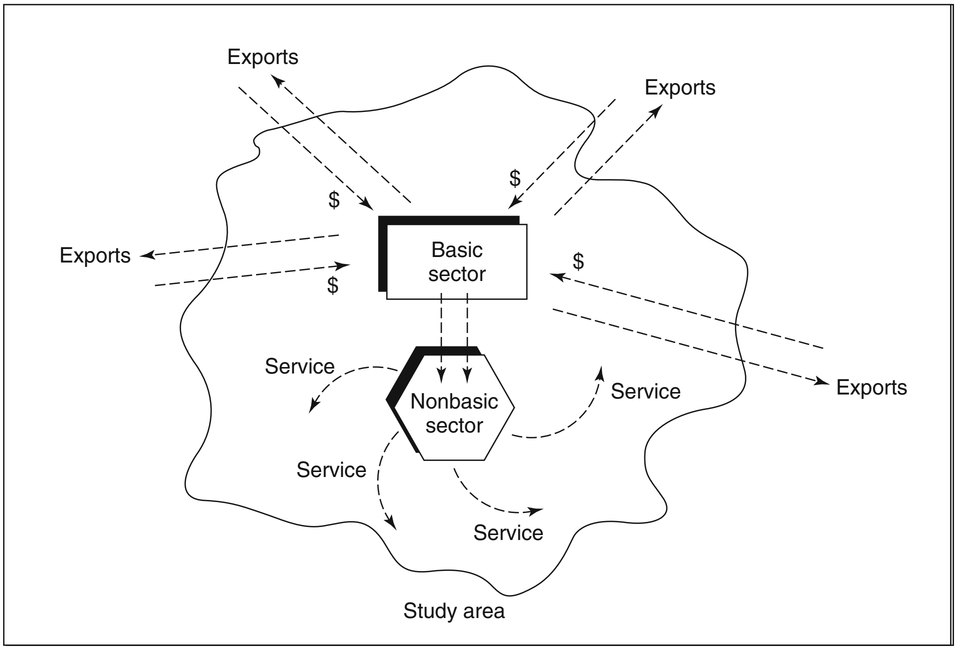
based on **time-series data**. Using cross-sectional data, models are calibrated on the current spatial distribution of activities, thus examining a "snapshot" of the population/employment distribution on the map. A time-series approach, on the other hand, would utilize not only the current pattern, but also previous patterns, which allows an observation over two or more time periods. The former is a static way of forecasting, while the latter is more dynamic. In other words, the former assumes the general activity distribution pattern will prevail over time, whereas the latter recognizes explicitly that changes over time are an integral part of the development. Aside from their important role in the development literature, the three economic concepts—economic-base theory, location theory, and input-output models—are selected for further discussion because the first two illustrate cross-sectional forecasting methodology, while the last one illustrates time-series forecasting.

A. Economic-Base Theory

The term **economic base** has many different usages and meanings so that it is necessary to clarify the definition for use here. In general, the term economic base has been applied to activities thought of as being major, fundamental, or of considerable importance in the economic structure of an area. The economic base of a community consists of those economic activities that are vital to the continued functioning and existence of that community. An economic-base study is an attempt to determine those economic activities devoted to the export of goods and services beyond the study area's borders. This activity is thought of as being the primary reason for the earning ability and economic growth of the community. Because these basic industries sell their products and services outside of the area, nonbasic or service industries can be supported within the community's boundaries. For example, barbers, dry cleaners, shoe repairers, grocery clerks, bakers, and movie operators serve others in the area who are engaged in the principal activities of the community, which may be mining, manufacturing, trade, or some other industry. These service industries have as their main function the provision of goods and services for persons living in the community.

This distinction of basic and nonbasic sectors of economic activity in an area is illustrated in Figure 2.1. Note that the income of the nonbasic sector is dependent upon the income of the basic sector so that it seems that the service industries only exist to serve basic workers and other service workers. Hence, fluctuations in income or employment in the basic sector will ultimately affect income and employment in the nonbasic sector. Since the nonbasic sector activities depend upon the basic sector, changes in the basic sector will have a net effect on the entire study area economy when some multiplier is applied to the economic-base method of analysis. The economic-base multiplier attempts to predict the change that will occur in the study area economy given a forecast of changes in certain basic activities. A significant part of the analysis involves the construction of these impact multipliers. They are numerical constants intended to impose the effects of changes in the demand for an area's goods and services upon the volume of employment or income in that region. For example, a government contract for a defense item increases employment in a firm by 2000 jobs. Indirectly both contract and job increases might generate still more work opportunities and produce a total increase in local employment two or more times a multiple of the original 2000.

Figure 2.1 CONCEPT OF BASIC VERSUS NONBASIC ACTIVITIES



SOURCE: Adapted from Newman (1972). Reprinted with permission.

Example
Using employment as the unit of measure, classify the employment of all industries in the study area as basic or nonbasic. Establish the Normal Ratio, the relationship between basic and nonbasic employment that usually exists:

$$\text{Normal Ratio} = \frac{\text{Nonbasic Employment}}{\text{Basic Employment}} \quad \text{(Assume a 2:1 normal ratio, for example.)}$$

Total Employment = Nonbasic Employment + Basic Employment. Assuming the total study area employment to be 90,000, then nonbasic employment is now 60,000 and basic employment is 30,000.

$$\text{Multiplier} = \frac{\text{Total Employment}}{\text{Basic Employment}} = \frac{60{,}000 + 30{,}000}{30{,}000} = 3$$

If basic employment is forecast to increase by 15,000, the total increase in nonbasic employment would be 3 × 15,000 = 45,000. Then the total employment for the forecast year becomes 15,000 + 45,000 + 90,000 = 150,000. Since the normal ratio of 2:1 still holds, nonbasic employment is 100,000 and basic employment is 50,000. ■

Thus, economic-base theory is to describe the development of economic activities in a typical area or region. The development of economic activities in a specific area can be explained in terms of the following four stages:

Step 1: Calculate the total population and employment and the amount of constituent basic and nonbasic (service) employment;

Step 2: Estimate the proportion of basic employment to population and that of population to service employment;

Step 3: Estimate the future trend in the basic employment; and

Step 4: Calculate the total employment and total future population on the basis of the future trend in basic employment.

In other words, basic employment has to be determined exogenously, then based on the multipliers such as labor force participation rate and population-serving ratio, which are the two proportions mentioned in Step 2, future employment and population in the region are estimated. Aside from the example above, another numerical example of the economic-base concept was given in Chapter 1 in Table 1.1.

The validity of future estimates of employment (or any other variables) depends upon the relative stability of the nonbasic-to-basic ratio developed. However, the economic-base method still has many problems to be solved. Some of these are:

1. Determining which activities are basic and nonbasic;
2. Choosing which units of measurement best represent the economy; and
3. Establishing the geographic area boundaries for which the base study is to be made.

In addition to these conceptual problems, other criticisms of the economic-base method have been registered. As the size of the study increases, the ratio of nonbasic to basic employees increases with a resultant increase in the multiplier. As a consequence, large areas have very large multipliers which do not truly reflect total economic change due to changes in the basic sector. It becomes apparent that the economic-base multiplier method is most applicable to relatively small areas and towns. Some critics challenge the premise that basic activities are more important than service activities because of the important contributions of such factors as the transportation system, communications network, and other systems serving the community. This criticism is important because planners use the basic-nonbasic distinction to emphasize which industries should be built up to improve the community's economy and to improve the balance of payments. Industries that produce goods which are presently imported would be neglected under this premise. More technical treatment of the subject will be found in Chapter 3.

B. Location Theory

Location theory, a study of the effects of space on the organization of economic activities, is a body of knowledge about the location of different activities or the rationing of different resources so as to achieve desirable spatial interaction. It has its genesis from early studies of the relative locations of plants and industry, in which the availability of raw material and the accessibility to consumer markets are of primary importance. According to the spatial price theory, transportation cost is the price for rationing resources and economic activities. For example, manufacturing plants and industries find the most convenient locations at close proximity to the input resources (both labor and raw materials) or consumer markets in order to minimize transportation costs. Another good example is a family's choice of housing location, in which a tradeoff is made between the transportation costs and other expenditures and values. If a heavy weight is placed on freedom from the noise and rush of the central city, the family locate at a distance away from the city and pay the transportation cost. In their decision, the utility of a serene environment is much higher than the utility of being close to jobs and other urban amenities.

One of the familiar location models is the **gravity model**, which states that the interaction between two subareas is proportional to their activity levels, but inversely related to their spatial separation. **Reilly's law of gravitational attraction**, for example, is based on the concept of spatial interaction. One of the first retail models was constructed out of this theory. This model uses the number of business activities, people, store sales, area, and so forth as an index of size and the fundamental measure of attractiveness of a central place. Consider a household located at *I′* choosing between the shopping centers at *A* and *B* as shown in Figure 2.2, or the reverse situation where a shopping center *I′* is to be located to serve the population at *A* and *B*. In general, the markets captured from *A* and *B* are in the ratio

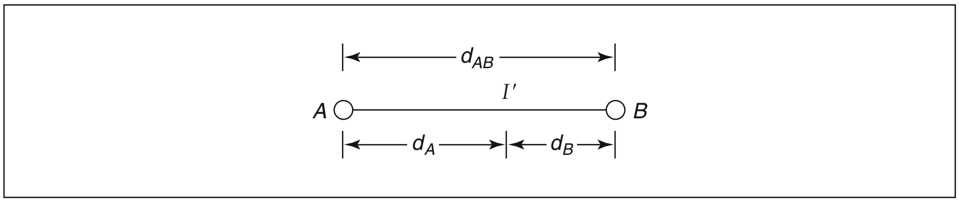
$$\frac{T_A'}{T_B'} = \frac{W_A}{W_B}\left(\frac{d_B}{d_A}\right)^2 \tag{2.1}$$

where W_A and W_B are the sizes of *A* and *B*, where T_A', T_B' represent proportions of trade (percentage of sales for example) from *I* to *A* and *B* respectively, and d_B, d_A is the distance from B and A respectively, with $d_A + d_B = d_{AB}$.

From Equation 2.1 attractiveness of *A* and *B* with respect to point *I′*, when *A* and *B* are of equal size ($W_A = W_B$), can be represented as $T_A' d_A^2 = T_B' d_B^2$. Notice the appeal of *A* and *B* is a function of both distance away and sales volume. To locate a shopping center at *I′* equally appealing to both the population centers *A* and *B*, or to say it the other way, to find the point *I′* where a shopper is indifferent between shopping centers *A* and *B*, we set $T_A' = T_B'$ in Equation 2.1 and solve for d_B. In general, an equation can be derived that states the watershed trade area bounded between *A* and *B*, measured in miles (km) from *B*, is

$$d_B = \frac{d_{AB}}{1 + (W_A / W_B)^{1/2}} \tag{2.2}$$

Figure 2.2 BREAK POINT MODEL



Example
Let d_{AB} = 36 miles (57.6 km); W_A = 92 retail activities, W_B = 90 retail activities; then d_B = 17.8 miles (28.5 km) from location B according to Equation 2.2. ■

The Reilly model may be an acceptable approximation for such location decisions in rural areas where central places are rather distinguishable. In a more developed area, however, a large number of shopping centers and population centers are involved. The overlapping market areas will be too complex to be resolved by this idealized model. Another formulation of the gravity model was proposed by Lakshmanan and Hansen (1965). This model allocates retail dollars, determining the percentage of the population in subarea i that will go to the shopping center j to spend their money:

$$(expenditure)_{ij} = (expenditure)_i \frac{W_j/\tau_{ij}^{\beta}}{\Sigma_k W_k/\tau_{ik}^{\beta}}$$

where τ is the travel time and β is the positive exponent to be calibrated. This states that the total consumer retail expenditure of population in subarea i is allocated toward each shopping center j in accordance with the gravity formula. Notice travel distance d is replaced by time τ in this formulation. We will see more of this interchangeability between time and distance in subsequent discussions throughout this book. **Huff's probabilistic model** (1962) is yet another example of the gravity model, stating that the probability a consumer located at i will visit shopping center j is

$$\frac{W_j/\tau_{ij}^{\beta}}{\Sigma_k W_k/\tau_{ik}^{\beta}} \tag{2.3}$$

Example
Suppose there are two shopping malls 5 and 10 miles (8 and 16 km) away respectively, each with 800 and 300 thousand square feet (72 and 27 thousand m^2) retail floor space. According to Huff's model, the probabilities a consumer will patronize these two malls are respectively

$$\begin{aligned} \frac{(800)(1/5^2)}{(800)(1/5^2)+(300)(1/10^2)} &= 0.08 \\ \frac{(300)(1/10^2)}{(800)(1/5^2)+(300)(1/10^2)} &= 0.92 \end{aligned} \tag{2.4}$$

assuming an exponent $\beta = 2$ (Dickey 1983). ■

Variants of location theory are found in literature on multicommodity flow as well as short-run and long-run equilibria of economic activities. **Multicommodity-flow models** describe the simultaneous allocation of population, employment, resources and finished products between places of supply and demand. In the short run, most economic activities, including the places of supply and demand, are fixed in location. In the long run, however, they could relocate themselves somewhere else corresponding to the rationing scheme of the spatial price system. Short- and long-run multicommodity flows are often modeled by a generalized version of the gravity model and optimization models—subjects covered in Chapter 4.

C. Input-Output Models

Input-output models, developed by Leontief (1953), will be introduced with respect to two particular applications: local-impact studies and interregional-flow studies. As an example, local-impact studies reveal the possible changes in a single region. Interregional-flow studies, on the other hand, are to show the structural relationship between regions. The effect of an autonomous shock—such as the precipitous injection of basic employment into the study area as mentioned in economic-base theory—may be traced to, and through, the region under consideration. An essential part of an input-output model is an input-output table, which documents a set of economic multipliers similar to those found in economic-base theory. The input-output table (matrix) eventually gives rise to a set of simultaneous equations with production (or technical) coefficients (the multipliers) and activity variables. The set of equations can trace out, on a multisectoral basis, the implication of introducing a new industry into the study area (the autonomous shock). For example, if a new tourist trade is introduced into the area as a way to boost the local economy, what would be the implications on the economic activities associated with tourism such as the associated retail and entertainment industries? The set of simultaneous equations merely chain-up the sequence of effects together in a mathematical formulation through the use of a table or matrix where the rows are inputs (e.g., tourists) and the columns are outputs (e.g., retail sales). It can be thought of as a huge revenue/expenditure accounting system. The revenue side of the balance sheet shows how the output for each industry is distributed, and the expenditure side records for each industry the distribution-of-inputs per unit-of-output from all industries.

An example of such an input-output matrix is shown in Table 2.1 (Chapin and Kaiser 1979). Shown for a single region, the table records horizontally the output for each particular sector of the economy measured in terms of receipts from sales (of goods or services) to every other sector. Thus sector 1 may be the tourist industry, sector 2 may be retail, sector 3 entertainment, and sector 4

Table 2.1 EXAMPLE INPUT-OUTPUT TABLE

	Tourism sector	**Retail sector**	**Entertainment sector**	**Household sector**	**Final demand**
Tourism sector	$30	$20	$30	$25	$105
Retail sector	60	20	80	30	190
Entertainment sector	10	40	60	50	160
Household sector	40	20	30	15	105
Charges against final demand	140	100	200	120	560

SOURCE: Chapin and Kaiser (1979). Reprinted with permission.

households. Households receive 25 million dollars during the current time period in wages as employees serving the tourist industry, the entertainment sector receives 30 million dollars from tourism, retail receives 20 million dollars, and the tourism sector spends 30 million on itself. Read vertically, the table shows input in terms of dollars spent on purchases in a particular sector from all other sectors. Thus local households as a whole spend 40 million dollars this time period on tourism, the entertainment industry spends 10 million dollars on the tourism industry as part of the intersectoral trade, and the retail industry spends 60 million dollars.

The final demand column records purchases by the tourism, retail, entertainment, and household sector—the dollar transactions after all intermediate processing and handling are completed. For example, tourists inject a total of 105 million dollars (first row sum) into the economy during this time period, divided among retail purchases, entertainment, and direct use of local labor. The charges against final demand in the bottom row are payments for tourism, retail trade, entertainment trade, and labor. Thus the fourth column (120 million) is the total wages paid to the household for supplying the labor for the remaining three sectors of the local economy, including the tourist industry, the third column is the total payment to the entertainment industry from other sectors and so on. These column totals are defined as the activity variables. To the extent that the row sums are not the same as column sums (or total purchases are not equal to payments) in Table 2.1, the final equilibrium values of these activities, taking the multiplier effects into account, are to be determined by the solution of a set of simultaneous equations.

From the dollar transactions in Table 2.1, production (or technical) coefficients are derived by dividing each input in a give column by the total of all inputs in the column. The resulting coefficients, shown in Table 2.2, are read by columns and indicate the cents-of-direct-inputs per dollar-of-output. Column 1 shows the input per dollar-value-of-output from each of all the other sectors supplying goods or services to sector 1. Thus the households contribute 29 cents toward the dollar on tourism, the entertainment sector contributes 7 cents, retail contributes 43 cents, and tourism pays itself 21 cents. The other columns show similar relationships for the retail, entertainment, and household sectors. The input-output technique, therefore, establishes a basic relationship between the volume output of any given industry in a region and the volume of input required in the production process from all other industries in this region. In this regard, the coefficients are equivalent to the labor force participation rate and population-serving ratio used in economic-base theory, except that the multipliers here are constructed out of dollar volumes rather than in terms of people. To the extent that intersectoral trade is governed by these multipliers aside from the seed activity (or autonomous

Table 2.2 PRODUCTION (TECHNICAL) COEFFICIENTS FOR A SINGLE REGION

	Tourism sector	**Retail sector**	**Entertainment sector**	**Household sector**
Tourism sector	0.21	0.20	0.15	0.21
Retail sector	0.43	0.20	0.40	0.25
Entertainment sector	0.07	0.40	0.30	0.42
Household sector	0.29	0.20	0.15	0.12

shock), the projection of the local economy, to be manifested in the final values of the activity variables, can only be determined following the four steps of economic-base theory, or alternatively solving the equivalent simultaneous equation set.

In the book, Chan (2005), more discussions of this Table can be found in the chapter on "Spatial Equilibrium and Disequilibrium." The similarity between input-output theory and economic-base theory will be emphasized. Most important, the input-output model will be extended from the current intraregional version to an interregional version.

II. ECONOMETRIC MODELING: INTERREGIONAL DEMOGRAPHIC PROJECTIONS

At the root of economic growth is population growth, for industrial wealth is nothing but a manifestation of human resources. An integral part of spatial economics is therefore the projection of population in a regional and interregional context. The demographic model is discussed here as a companion analysis to economic-base theory and input-output analysis. It also serves to illustrate economic theories, which are supplemental to classic economic theory in regional science. Three of the basic issues involved in demographic analyses are fertility, mortality, and migration. Fertility is the rate of childbirth in society. Mortality refers to the death rate in society. Migration is the population movement from one geographic location to another. Demographic analysis takes the net effect of fertility, mortality, and migration and predicts the growth or decline of population in the study area. The methods of analyzing demographic activities consist of population projection models, and matrix analyses of regional and interregional growth and distribution (Jha 1972). Population projection models are aggregate methods of extrapolating regional population growth from present trends using statistical techniques. The matrix analysis of population growth, on the other hand, is a more systemized method of projecting population growth, being more explanatory about the determinants of demographic activities.

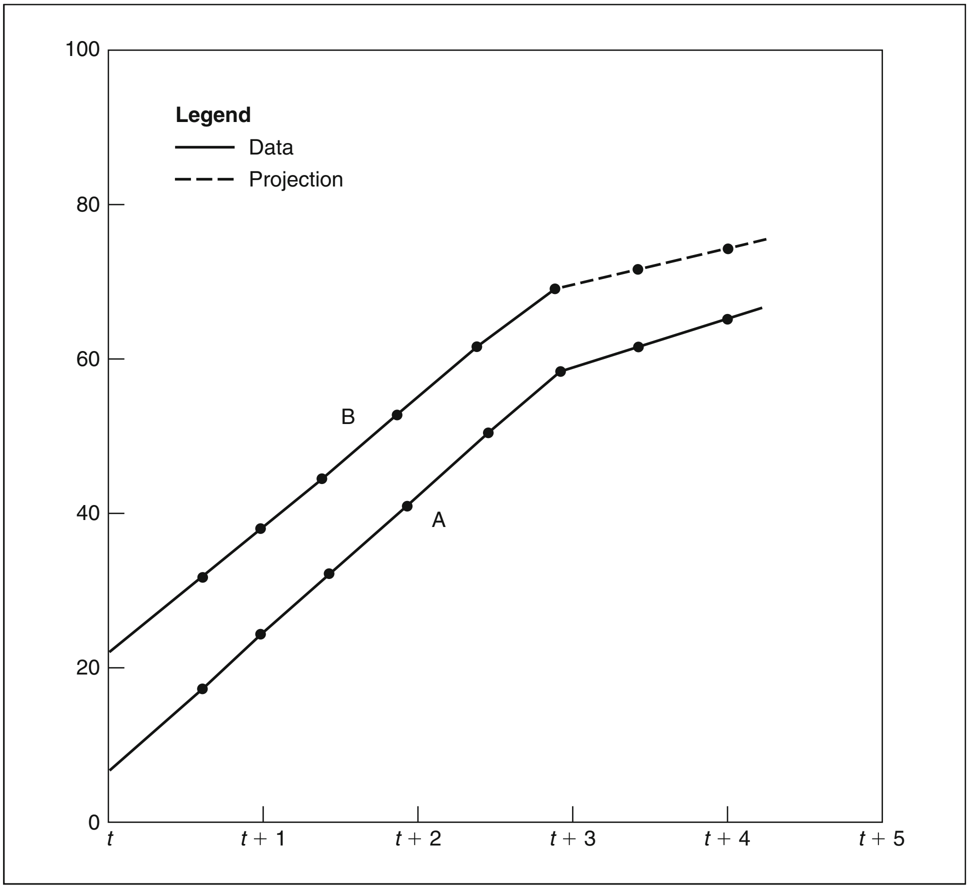
A. Population Projection Models

Two of the key concepts used in the population projection models are comparative forecasting and extrapolation. **Comparative forecasting** is a very crude method and could be rather unreliable if performed carelessly. This forecasting method is performed by selecting two areas, *A* and *B*, which have behaved similarly in their demographic growth patterns. It is assumed that the two areas should develop similarly in the future, meaning that if *A*'s population increases at a certain rate, *B*'s population would increase at about the same rate. Notice that *A* can be a part of *B* geographically. Parallel attempts are made to establish population and employment growth rate for similar cities. (See the "Econometric Models" chapter in Chan [2005]).

Example

As shown in Figure 2.3, if the population growth of two areas *A* and *B* are similar in the past from t to $t + 3$, and if the population of *A* is known for the rest of the years from time period $t + 4$ to $t + 5$, we can have an idea of the population projection for area *B* for the corresponding years. In this method, we assume that the

Figure 2.3 POPULATION PROJECTION BY COMPARATIVE FORECASTING

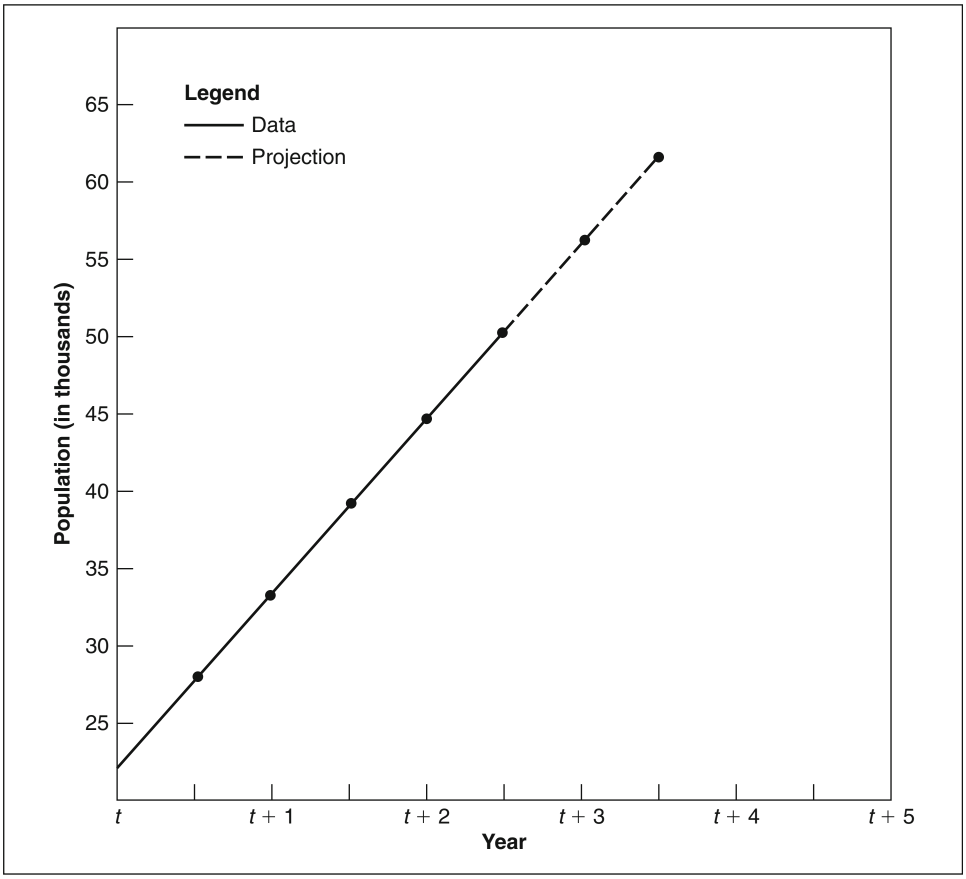


demographics of one area follow the same profile as the other. This will be true even if there is a sharp decline in growth rate occurring around time period $t + 3$. ■

Extrapolation, on the other hand, uses statistical techniques to predict future population growth based on the trend in the same area in the past. This is the basic premise of almost all econometric models, in which the implicit assumption is that past trends prevail. It represents both the strength as well as the weakness of this type of model. It is a strength since the forecasting methodology is flexible and relatively easy to use. It is a weakness inasmuch as the underlying behavior of the study area is ignored, in preference for purely statistical correlations. The common techniques employed in comparative and extrapolation models are graphical, polynomial curves, ratio and correlation method, regression and covariance method, and inflow-outflow analysis.

1. Graphical Method. The graphical or manual technique consists of plotting points on a graph to show population growth predictions. In this method, past census data is used for plotting the graph of population versus time. Future population is obtained by extending the graph in the same way as the trend in the past. Thus in Figure 2.4, the population at $t + 5$ and $t + 6$ have been obtained by

Figure 2.4 GRAPHICAL PROJECTION OF POPULATION AT REGION C



extending the graph. Simple as it may look, graphic plots of data are an essential, indispensable first step in any econometric application. They allow the modeler to get a feel of the data and more importantly to formulate a hypothesis about the structural form of the model. Pairwise plots such as those shown in Figure 2.4 are options in almost all statistical analysis software. Actual projection may not be actually performed manually, but the trend indicated by the plot is a most important piece of information for the modeler.

2. Polynomial Method. The polynomial-curve technique is a generalization of the above concept. It is built upon the following linearized formula for each forecast increment Δt: $N(t + \Delta t) + \delta N(t)\ \Delta t$, where $N(t)$ is the base-year population, Δt is the forecast period (whether it be one year, five years or ten years.), and $\delta N(t)$ is the population increase per time period Δt.

Example
If for an area, the total population in base-year t is 4500 thousand and the annual increment has been 27 thousand, then the population in $t + 10$ will be equal to $4500 + (27)(10) = 4770$ thousand. ∎

Polynomial curves are usually quite a bit more complex than the example shown above. For each time period, Δt, there exists a formal mathematical equation with a different increment as determined by the function $f(\Delta t)$: $N(t + \Delta t) = N(t) + f(\Delta t)$. Oftentimes, polynomial projections put more weight on present trends than past trends. One such weighting scheme is the **exponential smoothing technique** where the weight decays exponentially over the length of the elapsed time period, thus placing more value upon recent information. We will defer the details until the "Spatial Time-Series" chapter in Chan (2005), where formal projection methodologies will be discussed.

3. Ratio-and-Correlation Method. It might be possible that the population growth of the study area is related to the population growth of another area, or the region within which the area is located; or the population may be related to some socioeconomic factor such as employment of another area or the region. In this case, we use the ratio or coefficient of the relationship between the two areas for predicting future population, as shown in the following example.

Example
If the ratio of population at area A and any other socioeconomic factor at area B (including population) has been constant in the past years, then we can get the future area A population using this constant. Let $Z_B(t)$ represent the population or any other activity variable of area B at base year t, and suppose the ratio $N_A(t)/Z_B(t) = 0.8$.

If $$Z_B(t + \Delta t) = 4000 \text{ in the forecast year } t + \Delta t,$$

then $$\frac{N_A(t)}{Z_B(t)} = \frac{N_A(t + \Delta t)}{4000} = 0.8$$

$$\text{or } N_A(t + \Delta t) = (4000)(0.8) = 3200 \blacksquare$$

In other words, the ratio-and-correlation method uses another activity variable to predict population growth, if population growth can be correlated with an identifiable activity variable at a different area via a constant ratio. The reader can imagine that an example can easily be constructed for the interregional input-output model where the population in a region, being the support labor force for an industry, is simply related to the employment level at the work region by the labor-force-participation rate. The gist of this method is straightforward. If $N_i(t)/Z_j(t)$ = constant, then $N_i(t + \Delta t) = Z_j(t + \Delta t)$(constant). This model can be generalized to read

$$\frac{N_i(t + \Delta t)}{Z_j(t + \Delta t)} = f\left(\frac{N_i(t)}{Z_j(t)}, \frac{N_i(t - \Delta t)}{N_j(t - \Delta t)}, \ldots, \frac{N_i(t - n\Delta t)}{Z_j(t - n\Delta t)}\right) \tag{2.5}$$

where area i can also be area j ($i = j$), meaning that population and employment can be co-located in the same region. Here $f(\cdot)$ is a function showing how the constant can be determined by using historical information over n time periods. In the chapter on "Econometric Models" of Chan (2005), we will see how one can expand a great deal upon this very simple idea of ratio and correlation.

4. Regression and Covariance Analysis. This is one of the statistical calibration techniques widely used in population projection and for other activity variables as well. Here, population is taken as a dependent variable and another activity or factor is taken as an independent variable. Usually a simple bivariate regression may be represented like this: $N = a + bX$, where X is any explanatory or independent factor, a and b are calibration constants that may be obtained by fitting the model to the regional data. The companion covariance analysis, or analysis of variance, measures the quality of the statistical fit of the model to the data.

Example
If the population of a state is associated with the increase in per capita income X, and a and b have been calibrated to be 2,095,000 and 1,062 respectively. Further suppose that the forecast-year per capita income in the state is 15,000, then according to the regression equation above, future state population is projected to be (2,095,000) + (1,062)(15,000) = 36,880,000. ■

In general, while the regression equation does not necessarily have to be linear to start out with, it is often reduced to the following linear form before calibration can be performed: $N = a + b_1X_1 + b_2X_2 + \ldots$, where X_1, X_2 and so forth are independent variables. The regression coefficients b_1, b_2 and so forth are then calibrated for use in forecasting. Notice that the model assumes that the linear relationship between population and the independent variables will hold over time—very similar to the previous models, from comparative method to ratio-and-correlation method. The linearity assumption, and certain assumptions about the statistical distribution of the data, may impose restrictions on what is normally a very flexible modeling procedure. The technical aspects of regression and covariance analysis are discussed in Appendix 2 of this book.

5. Inflow-Outflow Analysis. The inflow-outflow analysis predicts the population of period $t + \Delta t$ into the future considering both the gain and loss of population in the area (termed **inflow** and **outflow** respectively.) The inflow is predicted by the equation

$$(\mathit{inflow}) = (\mathit{birthrate})\ N(t) + (\mathit{in\ migration})$$

The outflow, on the other hand, is predicted by

$$(\mathit{outflow}) = (\mathit{death\ rate})\ N(t) + (\mathit{out\ migration})$$

The population for the forecast year is predicted by combining the inflow and outflow results using the equation: $N(t + \Delta t) = N(t) + (\mathit{inflow} - \mathit{outflow})$. In summary, this method relates population projection to population growth, natural increase and decrease (due to birth and death respectively), and in-and-out migration via the following equation

$$N(t + \Delta t) = N(t) + \delta^N(\Delta t) + \delta^M(\Delta t)$$

where $\delta^N(\Delta t)$ is the natural increase or decrease in time period Δt, and $\delta^M(\Delta t)$ is the net migration during period Δt. Substituting and rearranging the terms, one can write $N(t + \Delta t) = N(t) + [b(\Delta t)N(t) + \delta N^I(\Delta t)] + [d(\Delta t)N(t) + \delta N^o(\Delta t)]$ where $b(\Delta t)$,

$d(\Delta t)$ are the birthrates and death rates during period Δt respectively, and $\delta N^r(\Delta t)$, $\delta N^I(\Delta t)$ are the in-and-out migrations during period Δt.

Example
If an area had a population of 4,500 for time period t, and the birthrate and death rate per capita are 2 and 1 percent respectively and the in-and-out migrations are 234 and 198 respectively for the forecast time increment, then the forecast population is 4,500 + (2(4,500/100) + 234) − (4,500/100 + 198) = 4,581. ■

B. Interregional Growth and Distribution

Matrix representation of population growth and distribution is convenient for estimating the growth patterns of multi-regional populations. Two methods will be introduced here: **cohort survival** and **components of change.** The cohort survival method is a way to determine population growth. **Cohort**, for this purpose, is defined as a group of people born within a given time period. The fundamental concept of this analysis is: $\mathbf{N}(t + \Delta t) = \mathbf{G}\ \mathbf{N}(t)$, where the population at a future period $\mathbf{N}(t + \Delta t)$ is related to the current period t via a matrix $\mathbf{G}$, the growth matrix. For analytical purposes, the population is broken down into cohort age groups. The matrix takes into account the death rates for each age group and incorporates them as survival ratio at the main diagonal of the matrix. On the other hand, the birthrates for each of the age groups are represented in the first row of the matrix. For example, the birthrate for age groups under childbearing age is zero, and similarly for those over the childbearing age. However, each group within the childbearing age would have a certain birthrate, suggesting their capacity to reproduce. The matrix determines the populations, by age group, for the forecast year based on survival and birthrates. The matrix also ages the base-year population into older groups for the forecast year. A group of residents in the five-to-ten-year age bracket, for example, would transition into the ten-to-fifteen-year bracket if the forecast is performed for a five-year increment. In summary, the following equation set incorporates all the above elements in a matrix notation.

$$\begin{pmatrix} N_1(t+\Delta t) \\ N_2(t+\Delta t) \\ N_3(t+\Delta t) \\ \cdot \\ \cdot \\ \cdot \\ N_n(t+\Delta t) \end{pmatrix} = \begin{bmatrix} 0 & 0 & b_3 & b_4 & \cdot & \cdot & \cdot & b_{n-1} & 0 \\ s_{12} & 0 & 0 & 0 & \cdot & \cdot & \cdot & 0 & 0 \\ 0 & s_{23} & 0 & 0 & \cdot & \cdot & \cdot & 0 & 0 \\ 0 & 0 & s_{34} & 0 & \cdot & \cdot & \cdot & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot & s_{n-1\,n} & 0 \end{bmatrix} \begin{pmatrix} N_1(t) \\ N_2(t) \\ N_3(t) \\ \cdot \\ \cdot \\ \cdot \\ N_n(t) \end{pmatrix} \tag{2.6}$$

where b_i stands for the birthrate per person for group i, and s_{ij} stands for the surviving ratio of group i in group j.

Aside from birth-death considerations, the problem of interregional migration can be taken into account by using a migration matrix. This matrix is similar to that used to model the survival rates of cohort groups, except that net immigration and emigration rates are written in the main diagonal. Since the matrix is used to model interregional population movement alone, no birthrates are included. In the following matrix, where the row and column dimensions correspond to the different age groups, net interregional population migration is modeled:

$$\begin{bmatrix} 0 & 0 & 0 & \cdot & \cdot & \cdot & 0 & 0 \\ m_{11} & 0 & 0 & \cdot & \cdot & \cdot & 0 & 0 \\ 0 & m_{23} & 0 & \cdot & \cdot & \cdot & 0 & 0 \\ 0 & 0 & m_{34} & \cdot & \cdot & \cdot & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdot & \cdot & \cdot & m_{a-1\,n} & 0 \end{bmatrix}$$

The growth of a region is predicted by adding the birthrate, survival-rate, and migration-rate matrices, which produces a growth-rate matrix by age group

$$\boldsymbol{G} = \begin{bmatrix} \leftarrow \bar{b} \rightarrow \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & \cdot & \cdot & 0 \\ s_{12} & 0 & \cdot & \cdot & 0 \\ 0 & s_{23} & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & \cdot & \cdot & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & \cdot & \cdot & 0 \\ m_{12} & 0 & \cdot & \cdot & 0 \\ 0 & m_{23} & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & 0 \end{bmatrix} \tag{2.7}$$

Example

A simple numerical example would illustrate these matrices. Consider three age groups: 0- to 20-year-olds, 20- to 40-year-olds and 40- to 60-year-olds. These hypothetical matrices can be written:

$$\boldsymbol{G} = \begin{bmatrix} 0 & 1.5 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0.9 & 0 & 0 \\ 0 & 0.8 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \end{bmatrix} \tag{2.8}$$

where the childbearing cohort group is defined as those 20 to 40 years old. We specify that 9 out of 10 people survive from the 0- to 20-year group to become 20- to 40-year-old adults. Ten percent more people in the 20- to 40-year-old group migrate into the area over 20 years—the length of the forecast period—and so on. Summing these matrices, we have the net growth matrix

$$\boldsymbol{G} = \begin{bmatrix} 0 & 1.5 & 0 \\ 1.0 & 0 & 0 \\ 0 & 0.9 & 0 \end{bmatrix}$$

If the base-year population in all age groups is 10,000, the forecast population distribution (in thousands) would be

$$\begin{pmatrix} N_1(t + \Delta t) \\ N_2(t + \Delta t) \\ N_3(t + \Delta t) \end{pmatrix} = \begin{bmatrix} 0 & 1.5 & 0 \\ 1.0 & 0 & 0 \\ 0 & 0.9 & 0 \end{bmatrix} \begin{pmatrix} 10 \\ 10 \\ 10 \end{pmatrix} = \begin{pmatrix} 15 \\ 10 \\ 9 \end{pmatrix} \tag{2.9}$$

It is predicted, therefore, that in 20 years more young people than older people will be living in the study area. More precisely, there will be 15 thousand 0- to 20-year-olds, 10 thousand 20- to 40-year-olds, and only 9 thousand 40- to 60-year-olds. ■

C. Interregional Components of Change Model

Predicting interregional population is basically the same as predicting regional population. The major differences are that instead of breaking down by age groups, we stratify by specific regions, such as the East versus West Coast. This basic concept is still used:

$$N(t + \Delta t) = N(t) + (births) - (deaths) + (migrants)$$

Symbolically, the components of change model may be stated in scalar terms for each region *i* as

$$\begin{aligned} N_i(t + \Delta t) &= N_i(t) + b_i(t)N_i(t) - d_i(t)N_i(t) + m_i(t)N_i(t) \\ &= [1 + b_i(t) - d_i(t) + m_i(t)]\, N_i(t) \\ &= g_i N_i(t) \end{aligned} \tag{2.10}$$

where *b*, *d*, and *m* are birth-, death and net migration rates. For example, the crude birth-, death and net migration rates from Table 2.3 give rise to the growth rate $g = 1 + 0.1315 - 0.0473 + 0.0865 = 1.1707$. These are called crude because they are simply the births, deaths, and net migration over the period 1955–60 divided by the 1955 base-year population in California, without taking into consideration migration from/to the rest of the United States or any place else. In fact, proper estimation of these parameters is a subject of interest in real world applications. Chan (2005) elaborates on this topic in the "Bifurcation and Disaggregation" chapter. (Software and data, under the YI-CHAN folder, are also included on the CD/DVD attached to this book to illustrate the estimation procedure.) Usually, population, births, deaths, and migration are expressed in matrix forms, where the row and column dimensions correspond to the number of regions being modeled. The following model shows a two-region example in which the internal births, deaths, and interregional net migration are analyzed.

$$\begin{pmatrix} N_1(t + \Delta t) \\ N_2(t + \Delta t) \end{pmatrix} = \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} b_1(t) & 0 \\ 0 & b_2(t) \end{bmatrix} - \begin{bmatrix} d_1(t) & 0 \\ 0 & d_2(t) \end{bmatrix} + \begin{bmatrix} 0 & m_{21}(t) \\ m_{12}(t) & 0 \end{bmatrix} \right\} \begin{pmatrix} N_1(t) \\ N_2(t) \end{pmatrix} \tag{2.11}$$

or in matrix notation $\boldsymbol{N}(t + \Delta t) = (\boldsymbol{I} + \boldsymbol{B} - \boldsymbol{D} + \boldsymbol{M})\, \boldsymbol{N}(t) = \boldsymbol{G}\, \boldsymbol{N}(t)$.

Table 2.3 CALIFORNIA AND THE REST OF THE UNITED STATES (1955–60)

Region	1955 Pop	Birthrate	Death rate	Migration rate
Calif	12,988,000	0.1315	0.0473	0.0865 (~US to Calif)
Rest of the US (~US)	152,082,000	0.1282	0.0488	−0.0074 (Calif to ~US)

Example
From the data in Table 2.3, the growth matrix is the sum of the identity, birth, death, and migration matrices, where California is row/column 1 and the rest of the United States is row/column 2 of such matrices:

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0.1315 & 0 \\ 0 & 0.1282 \end{bmatrix} - \begin{bmatrix} 0.0473 & 0 \\ 0 & 0.0488 \end{bmatrix} +$$

$$\begin{bmatrix} 0 & 0.0865 \\ -0.0074 & 0 \end{bmatrix} = \begin{bmatrix} 1.0842 & 0.0865 \\ -0.0074 & 1.0794 \end{bmatrix} \qquad (2.12)$$

The 1960 population in California and the rest of the United States can then be computed as

$$\begin{pmatrix} \mathrm{N}_1(1960) \\ \mathrm{N}_2(1960) \end{pmatrix} = \begin{bmatrix} 1.0842 & 0.0865 \\ -0.0074 & 1.0794 \end{bmatrix} \begin{pmatrix} 12{,}988 \\ 152{,}082 \end{pmatrix} = \begin{pmatrix} 27{,}236 \\ 164{,}061 \end{pmatrix} \blacksquare \qquad (2.13)$$

The discussion on interregional demographic model gives the reader a flavor of the basic algebra found in similar model structures as the interregional input-output model. It serves not only to introduce econometric modeling, but also to generalize to a multi-regional level a key projection concept introduced earlier in this chapter.

III. ECONOMIC CONSTRUCTS FOR COST-BENEFIT ESTIMATION

The previous sections have been devoted to the economic and econometric techniques of prediction where future activities, such as the local economy, are projected. In this section, we will concentrate on the methods of evaluation, in which a location or land use policy is analyzed or evaluated with respect to its cost and benefits. There are three economic concepts that are important to cost-benefit estimation: equity, efficiency, and externality. **Equity** is a very precise concept in economics since it connotes the distribution of income and social benefits. An example may be the equal accessibility of all segments of the population to such public services as school and recreation (Marsh and Schilling 1994). Equity can be achieved through the natural market forces, governmental intervention, or through public services and transfer payments. The price system may sometimes be inadequate to effect an equitable distribution of goods and services; it may then be necessary to subsidize schools in a less affluent neighborhood in order to render education opportunities for all.

Efficiency, in our context, means the least costly distribution of resources over space for the production of goods and services. An efficient urban structure, for example, is to have complementary goods and services to be clustered together, whereby transportation costs are minimized.[1] Such a clustered development may mean the sacrifice of some open space that is sometimes highly valued. Efficiency, therefore, is not necessarily the only objective of urban planning; other factors need to be considered at the same time.

Externality, for the purpose of the current discussion, refers to the effects of a project other than those measured by the economic price system. In the provision of open space above, transportation cost does not accurately represent the price for the distribution of open space around the city, meaning that a precise, quantifiable price measure of the value of open space to an inhabitant is not easily obtainable. Economists have a well-defined concept about price theory, and they recognize that certain effects cannot be measured by price, including the positive benefits of open space and the negative benefits of air pollution. However, in a comprehensive accounting system, we may like to impute a cost to the community for the deprivation of open space, or the onset of pollution, both of which may be incurred in the industrial production process. This imputed cost is an example of an externality (Dahlman 1988).

Having been equipped with these basic concepts, we are prepared to examine two sets of methodologies for estimating costs and benefits. The first is **shift-share analysis**, which illustrates a technique to measure equity in a spatial context. The second is **theory of land values**, which is included here to verify the concept of efficiency.

A. Shift-Share Analysis

Shift-share analysis is a technique to divide the change in a socioeconomic measure into two or more components. For example, the population growth in an area is attributable to both the regional growth pattern and the peculiarity of the area itself. This technique can be used to measure the distribution of benefits: for instance, which subarea in the study area will receive less than its equitable share of regional growth and which will receive more. Rather than assuming a constant trend and a constant share of the regional economic activities, shift-share analysis tries to explain the change in the activity level in a particular subarea by two components. The first component is an average activity change corresponding to an aggregate regional change, while the second component is the difference between the average and actual changes in a subarea. This can be expressed by the following equation:

(*subareal change*) = (*regional average change*) + (*competitive change*)

For example, an urban area grows 10 percent over a five-year period, and two of its zones *A* and *B* grow by five percent and 12 percent respectively. Zone *A* is at a competitive disadvantage of five percent below while zone *B* is at an advantage of two percent above the regional average, even though both are influenced by the overall regional growth.

A general expression of shift-share analysis can be written for activity *k* in subarea *i:*

$$\Delta Z_i^k = \delta Z^k + \delta Z_i^k = \frac{\Delta Z^k}{Z^k(t)} Z_i^k(t) + \delta Z_i^k \tag{2.14}$$

which states that the total change of activity *k* in subarea *i* is due to subareal change of activity *k* at the regional rate, adjusted for site-specific change at the local level. Shift-share analysis is therefore a simple concept of splitting up the change in activity from time period *t* to *t* + 1 into two functional components.

The first component indicates the norm for the region as a whole and the second the subareal deviation from the norm as mentioned. Notice that the competitive component is introduced to measure the change in a subarea relative to the regional average—showing the relative attractiveness of the subarea for the particular activity under consideration.

The competitive component of change in activity k for subarea i, δZ^k_i, can again be broken down into two components: the difference between subareal change and the regional overall growth in sector k.

$$\delta Z^k_i = Z^k_i(\mathrm{t})\left[\frac{\Delta Z^k_i}{Z^k_i(t)} - \frac{\Delta Z^k}{Z^k(t)}\right] \tag{2.15}$$

Putting it altogether, we can see that δZ^k in Equation 2.14 defines the change in importance of industrial sector k in subarea i over the time period, or the shift component. Equation 2.15, on the other hand, defines the increase or decrease in activity k due to the relative competitiveness of subarea i vis-a-vis other subareas, or the share component. This accounts for the name shift-share analysis.

Example

During the past five years, subarea i's manufacturing (M) sector grew less rapidly than did the region by 1.6 percent. Its commercial (C) sector, in contrast, had a growth rate that exceeded that of the region's by 3.8 percent. Regional manufacturing and commercial growth rates are given as 0.276 and 0.402 respectively (i.e., 27.6 percent and 40.2 percent), and the current subareal manufacturing and commercial activity levels are $280,000 and $180,000 respectively. Assuming a constant shift, what is the value of manufacturing and commercial trade in a projected time period?

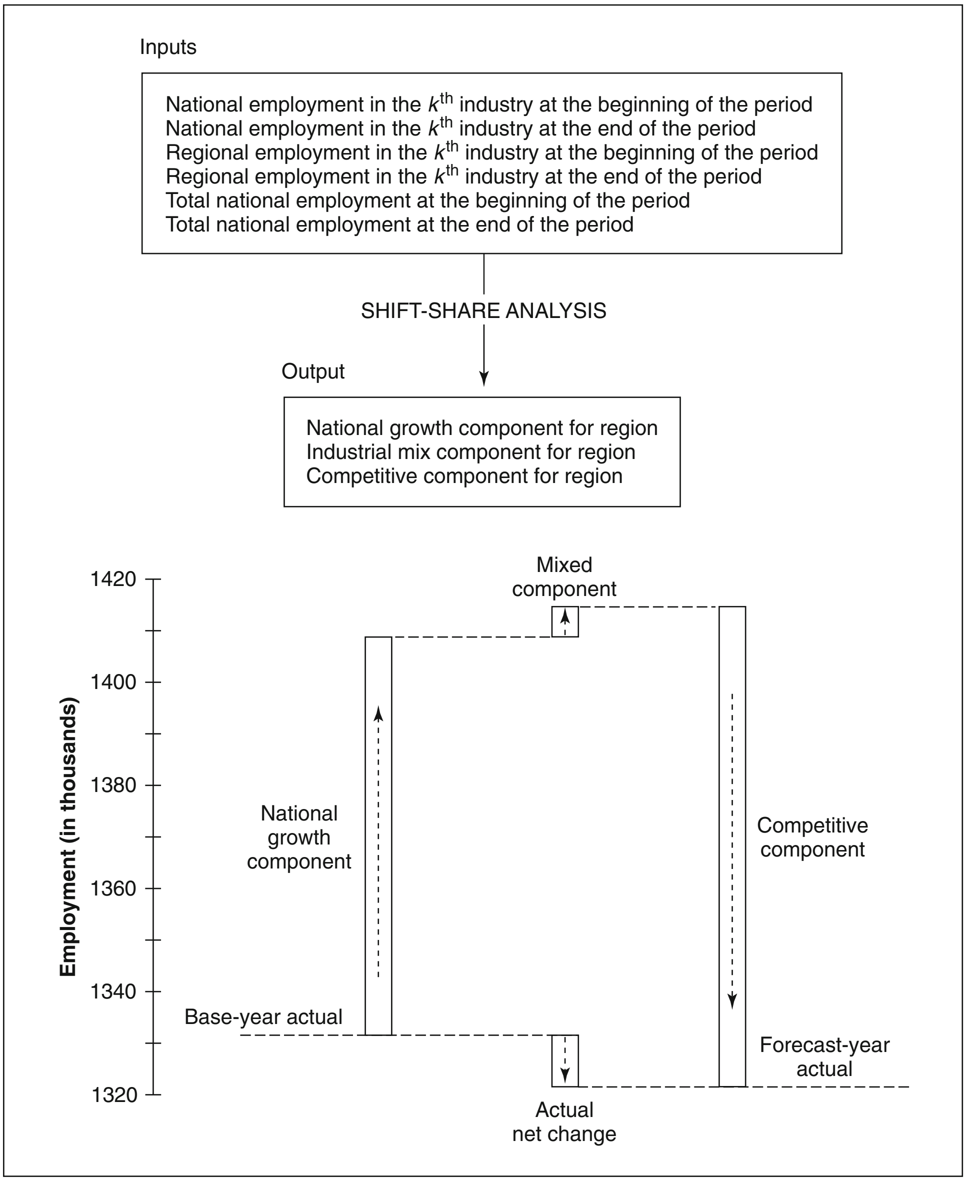
To answer this question, we add the national growth rate to the subarea's growth rate and multiply the result by the subarea's current sectoral activity level according to Equation 2.15, yielding the projected manufacturing and commercial levels as requested:

$$\begin{aligned} \delta Z^M_i &= (-0.016 + 0.276)280 = 72.8 \\ \delta Z^C_i &= (+0.038 + 0.402)180 = 79.2 \end{aligned} \tag{2.16}$$

In this shift-share example, the first term in Equation 2.14 disappears since we assumed constant shift (Krueckeberg and Silver 1974). ■

Figure 2.5 illustrates another example in the relationship between a regional economy and the national economy where all three components are present: national growth component, industrial mix component, and the competitive component. It shows the input data required to estimate each of these components, as well as a graphic plot of a numerical example for regional employment. Thus the drop in regional employment from 1332 to 1321 thousand is explained in terms of these components. The concepts presented in shift-share analysis, while simple, are not readily used in the field, since we never discussed how the growth rates are actually derived beyond the schematic as illustrated. Chan (2005) shows in his "Spatial Equilibrium and Disequilibrium" chapter that implementation potentials can be enhanced by including this concept within the interregional version of input-output analysis.

Figure 2.5 EXAMPLE APPLICATION OF SHIFT-SHARE ANALYSIS


Inputs
National employment in the k^{th} industry at the beginning of the period
National employment in the k^{th} industry at the end of the period
Regional employment in the k^{th} industry at the beginning of the period
Regional employment in the k^{th} industry at the end of the period
Total national employment at the beginning of the period
Total national employment at the end of the period
SHIFT-SHARE ANALYSIS
Output
National growth component for region
Industrial mix component for region
Competitive component for region
Employment (in thousands)
1420
1400
1380
1360
1340
1320
Mixed component
National growth component
Base-year actual
Competitive component
Forecast-year actual
Actual net change

B. Theory of Land Values

Having completed our discussions on equity measurement, let us now turn to the concept of efficiency and illustrate it through the theory of land values. Land value is subject to the market forces of supply and demand and highly related to location and transportation costs. An improvement of the transportation system, such as new highway or subway construction, could affect land value significantly. Dorau and Hinman, as far back as 1928, suggested tracing land value to three additional explanatory variables: land income, rate of capitalization, and direct satisfaction from land ownership. Land income includes mortgages as well as the rent collected from tenants on the property, and in general the usefulness of the land corresponding to the various services it can render. While it may be obvious that land value depends on how potential income can be obtained from the land, it needs to be pointed out that such income includes not only those from the current time period, but also the forthcoming periods. This means that the rate of capitalization, such as interest rates, risk, and other investment preferences, are involved. The last explanatory factor—direct satisfaction from ownership—needs little explanation. It pertains to the personal rewards that are not measured by the monetary system.

Thus it can be seen that in a cost-benefit analysis, if land value is the primary measure of benefit, there are a variety of means to effect the change in land value, each of which would probably incur a cost. Improving accessibility by building highways, for instance, is a way among many others. The theory of land values helps to explain such a cost-benefit relationship, and in a practical sense, contributes toward model building. Aside from the above observations, there are several economic phenomena that are useful for model building as well. It is observed, for example, that land value or land rent declines with the distance from the central business district. The further one goes away from the central city, the lower the land value. Land rent and transportation costs are complementary. Thus in a hypothetical, circular city, the land values can be viewed as a cone in three dimensions (see Figure 2.6). If one wishes to live in the central city, the land rent is at a peak, but the transportation costs are at a minimum. On the other hand, if one locates at the fringe of the city, the land rent will be low, but the transportation cost will be high. You can either pay a high rent and be accessible, or you can pay a low rent and be comparatively inaccessible, hence having to pay more on transportation costs. Land rent is affected by transportation in another way. In the case of Philadelphia and other cities with a radial highway system, the development follows along the freeways in a finger-like manner. Suppose one adapts Burgess's classic concentric zone structure to an urban area consisting of contours of land value in rings around the city center. After a freeway is built, the development would tend to align itself along the freeway, stretching out the rings as indicated in Figure 2.7. In this case, Burgess's theory merges with Hoyt's sector theory,[2] which suggests that there are modifications to the Burgess's concentric rings to reflect transportation corridors that induce suburban development along the corridors.

Figure 2.6 LAND RENT AND TRANSPORTATION COST

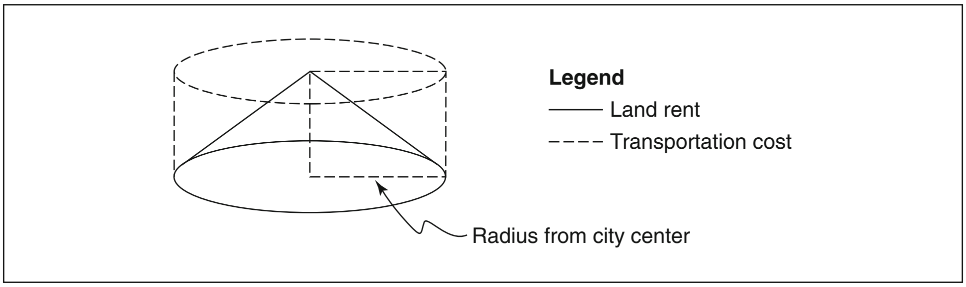
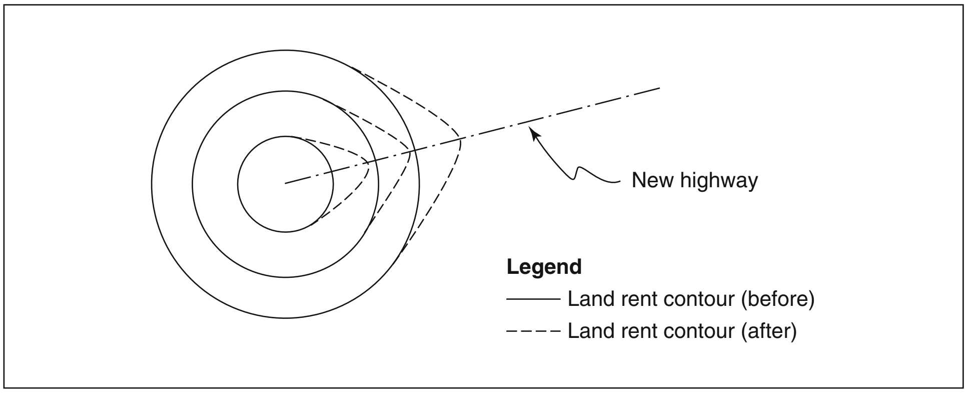


Figure 2.7 EFFECT OF TRANSPORTATION ON LAND USE


New highway
Legend
——— Land rent contour (before)
- - - - Land rent contour (after)

C. Consumers' Surplus

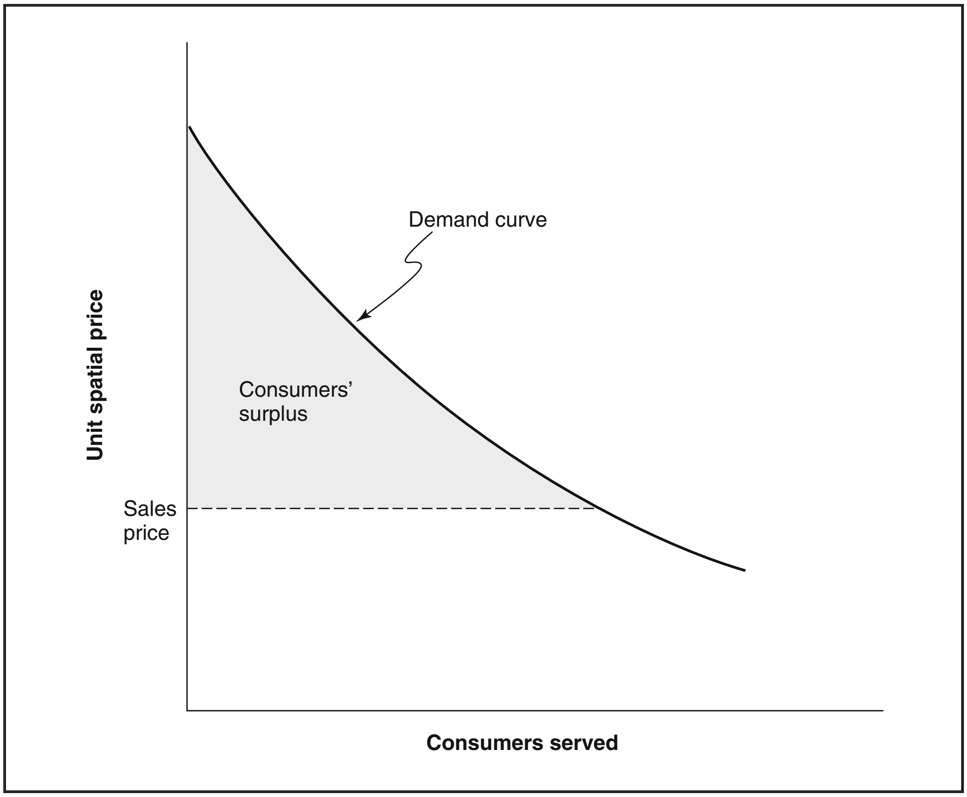
When economic efficiency is of concern, a valuation measure in spatial choice is **consumers' surplus**. The consumers' surplus is defined as the difference between what consumers might be willing to pay for a location and what they actually pay. As shown in Figure 2.8, the consumers' surplus is the area between the demand curve and the spatial price. Since the demand function expresses the users' indifference between the utility of a location and money, it can be considered as an expression of the utility of locations in terms of prices. The consumers' surplus, which is expressed in monetary units, is then a measure of the utility provided to the consumer minus the cost of production, which is reflected in the sale price to some degree. Maximization of consumers' surplus is then a close proxy of the maximization of the economic utility of the consumers. The evaluation of projects through a consumers' surplus analysis is widely, although generally only implicitly, used for large-scale public facilities. It is the only effective means of estimating economic benefits when the public facilities are so large as to effect more than marginal changes in prices.

To estimate the change in consumers' surplus brought about by any project, it is necessary to know both the price and the scale of the facility built before and after the project is completed. Figure 2.9 shows the change in consumers' surplus before and after a facility expansion from $\overline{P}_{bef}$ to $\overline{P}_{aft}$, which increases the number of consumers served from V_{bef} to V_{aft}. Algebraically, this change can be approximated by the trapezoid rule:

$$\frac{1}{2}(C_{bef} - C_{aft})(V_{bef} + V_{aft}) \tag{2.17}$$

Measurement of the equilibrium price C can be difficult when the project is large enough to shift the demand curve by causing an income effect. Such an income effect is illustrated in Figure 2.10, where the tradeoff between housing and transportation is considered.[3] The effective increase in income caused by a price reduction on a major facility shifts the point of maximum utility from U^*_{bef} to U^*_{aft}. The increase in income thus results in an increased demand for both transportation and housing. The income effect of a price change is only significant when major

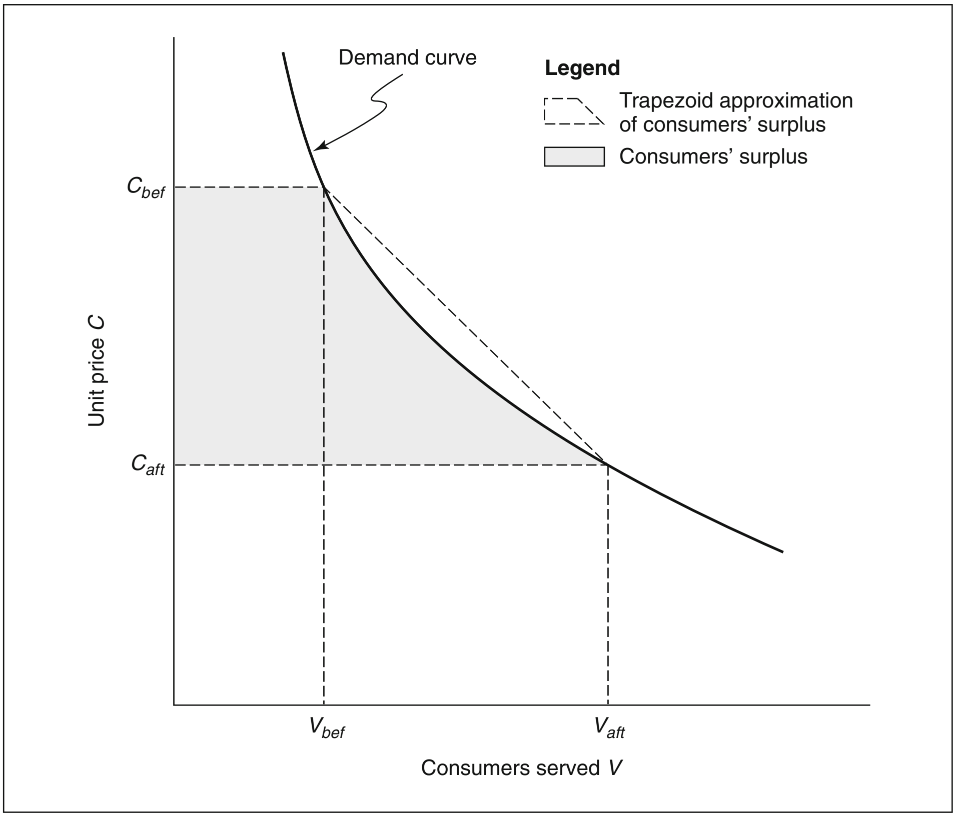
Figure 2.8 CONSUMERS' SURPLUS ILLUSTRATION



expenditure items are involved. For most families in the United States, these would be transportation, housing etc. Price changes on these items can change the level of consumption. Increased rent or housing costs could, for example, decrease the demand for travel. In developing countries, investments in basic infrastructure such as transportation, housing, and power can, by decreasing the cost of these items, significantly increase the effective income (I') of the inhabitants.

When income effect is involved, knowledge of the income elasticity of demand $\left(\frac{dV}{V}\Big/\frac{dI'}{I'}\right)$ is required in order to estimate the final price C_{aft} along the same demand curve. Equation 2.17 still provides a satisfactory, although more approximate, means of calculating consumers' surplus. Chan (2005) illustrates this calculation in his "Including Generation and Distribution" chapter, where he estimates the economic value of state parks. (The software that performs such calculation is included on the attached CD/DVD under the STATEPRK folder.) In calculating consumers' surplus, the analyst must be careful to reckon with the effects of manipulations of the prices through a deliberated pricing policy. In systems that are publicly owned, it is possible and sometimes desirable to set prices that cover more or less than the total costs. Hydroelectric power in the western United States, for example, was subsidized below average cost to promote development. Unless the subsidies are deducted, this policy clearly increases consumers' surplus over what it might be if full cost of the service were charged. Figure 2.11 shows the total consumers' surplus made up of that part by the market mechanism and the other

Figure 2.9 CHANGE IN CONSUMERS' SURPLUS



part by regulation. Such changes in consumers' surplus, effected by setting the prices of services different from their costs, are not without expenditure. The changes are indeed transfer payments that must be made up by subsidy, either from taxes or from profits in some other part of the system and deducted from the final consumers' surplus calculations of the project.

IV. UTILITY THEORY

Utility theory is a common economic concept to explain location choice and decision among alternatives in general. A view of utility functions may be developed in the following way. Each household is confronted with a choice between n different expenditures, including savings or dis-savings, within an income budget. This can be expressed by the following equation where p_i and x_i refer to the price and quantity of the i^{th} expenditure: $I' = \Sigma_{i=1}^{n} p_i x$. On the other hand, the household derives a certain amount of satisfaction from the quantities of each commodity it purchases, and this degree of satisfaction, when added up, provides a total utility. This utility may be expressed as a function of the vector of purchases of

Figure 2.10 THE INCOME EFFECT

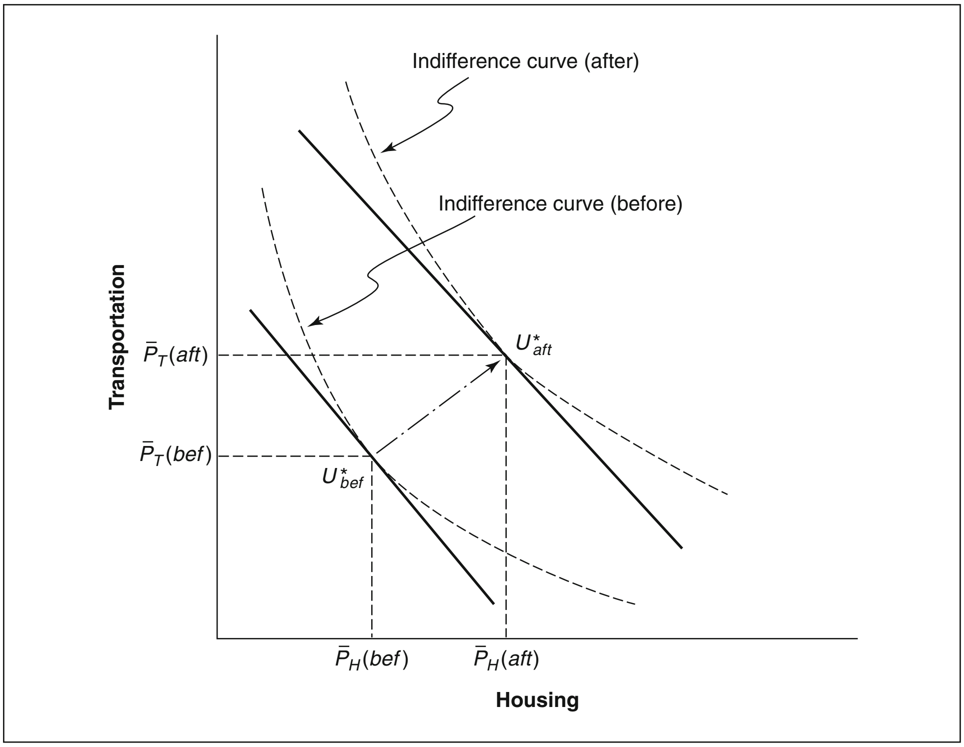
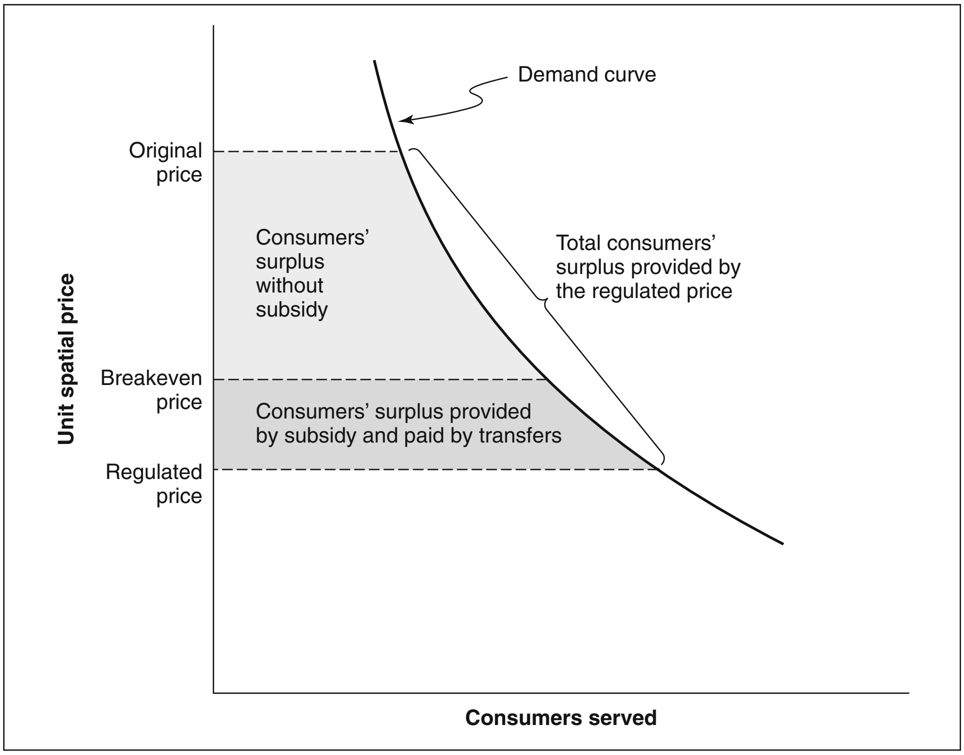


Figure 2.11 SUBSIDY AND TRANSFER PAYMENT


Demand curve
Original price
Consumers' surplus without subsidy
Total consumers' surplus provided by the regulated price
Breakeven price
Consumers' surplus provided by subsidy and paid by transfers
Regulated price
Unit spatial price
Consumers served

commodities and services **x**: $v = f(\mathbf{x})$; but this expression is vacuous until we specify the form of the function $f(\mathbf{x})$. One may assume, for example, that it could be linear:

$$v = \sum_{i=1}^{n} w_i x_i \tag{2.18}$$

This says that utility is the weighted sum of the purchases. This turns out to be not a very satisfactory idea because if a household tried to maximize its utility under this simple form, the whole budget would be spent on the commodity or service for which w_i/p_i was a maximum. Thus if the weight on travel was high and transportation cost was low, a family might spend its entire income on travel, which is somewhat absurd.

It would not help very much if we retain the linear model of Equation 2.18, but placed a requirement on the minimum consumption of each x_i. This would result in every commodity being consumed at its minimum level with the exception of the most cost effective one. A more complicated model can easily be devised in which various needs are each satisfied by a linear combination of commodities, and minimum values are set for the satisfaction of each need. This model is still unrealistic in that the minimum level of needs has to be set exogenously. Normally within the household, choices are made between the levels of satisfaction of various broad classes of needs—the need for housing, accessibility, non-housing, and non-location goods and services. Any linear model would force us to make decisions about these tradeoffs outside the model.

What makes tradeoff and consumption both possible and necessary is the fact that, for most goods, increasing quantities provide increasing satisfaction, but at a decreasing rate. Thus if twice the space is available to a household by moving further away from the city, the increased space may not double the housing satisfaction. In some cases, it might even decrease it. If we assume that increasing amounts of a commodity always add something to a household's utility, or at least never subtract from it. Suppose we also assume that the increase in satisfaction for each additional unit of a given commodity is diminishing, we have familiar economic statements about utility functions which are usually expressed mathematically:

$$\begin{aligned} \partial v/\partial x_i &\geq 0 \qquad i = 1, 2, \ldots, n \\ \partial^2 v/\partial x_i^2 &\leq 0 \qquad i = 1, 2, \ldots, n \end{aligned} \tag{2.19}$$

An example function is

$$v = \sum\nolimits_i a_i \ln x_i \tag{2.20}$$

or alternatively

$$v = \prod_{i=1}^{n} a_i x_i \tag{2.21}$$

A form of the utility function corresponding to these two is extremely useful for our discussion here because we are dealing with commodities which are, in the western culture, absolutely essential. Every family must have housing, access to employment, and other commodities such as food and clothing. If one of these commodities is reduced to zero in Equation 2.21, the level of utility falls to zero. A utility function of this type leads to tradeoffs that give adequate weight to extreme deprivation of any of the essential commodities of life. While Equations 2.20 and 2.21 are useful utility-function forms, alternative approaches exist to quantify a decision maker's values. In Chapter 5 the multi-attribute utility theory will be introduced, which is based more on behavioral grounds.

A. Estimating Bid-Rent via Utility Function

Before utility can be measured, the terms of the utility function must be defined. Part of the satisfaction from a particular residential location may be associated with the accessibility to work and/or recreational facilities in an area. Another may be connected to the availability of schools or pleasantness and quiet of the community. Let us now see how these are actually being quantified. First, we stratify the population by income, family size, and other socioeconomic factors, not only to detect different behaviors, but also to be sure that we are dealing with relatively uniform levels of housing and related expenditures. In the discussion that follows, it should be understood that income is fixed at a class mean, or at least falls within a relatively narrow range as a result of the stratification of individual households.

Alonso (1970) has the idea of measuring utility with reference to income, whereby the utility function takes into consideration the total available income. In a family's budget, let us define M' as the non-location expenditures, which include items such as food, clothing, and education. M' also includes savings at a bank. Another expenditure is rent (r), which includes mortgage payments, rent, and utility bills. Then we have transportation cost represented by T. Collectively r and T are referred to as location expenditures. These budget components can be broken down further, but the way we are doing it now satisfies our purpose. All these expenditures must fit into the budget I': $I' = M' + r + T$, which says certain parts of the income go to location and another to non-location expenditures. The simple equation above also underlines the complementary relationship between transportation outlay and rent, as covered earlier in this chapter when we discussed land rent theory.

We will now assume a particularly simple form of the utility function referenced as Equation 2.20:

$$v = \ln M' + \alpha_1 \ln H + \alpha_2 \ln A + \alpha_3 \ln C' \tag{2.22}$$

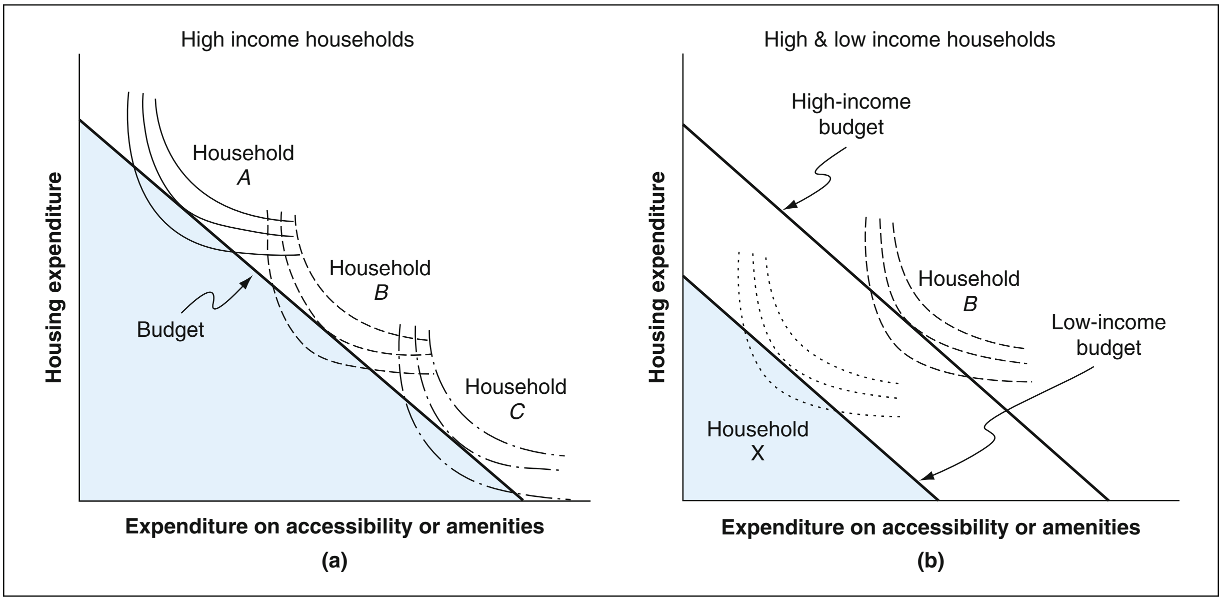
Here, M' stands for the consumption of all non-location goods as discussed above, while H, A, and C' stand respectively for the expenditure on providing housing, accessibility, and community amenities. In Equation 2.22, α_1, α_2, and α_3 are coefficients defining the relative importance of housing, accessibility, and amenities. We now introduce a basic assumption of overriding importance, whose application to this problem is due to Alonso (1964, 1970). We assume that for a particular set of households of homogeneous tastes, utility is uniform wherever they are located in the metropolitan area. We cannot, of course, be sure that by defining homogeneous socioeconomic groups, we have actually defined groups whose preferences in the housing market are also homogeneous. Given some uniformity in tastes, however, the assumption of equal utility is based on elementary economic considerations. If the utilities being enjoyed are in fact not equal and if there are locations in which a particular group could enjoy a higher utility, members of that group will bid up the price of land and housing at that location. The higher cost of the housing package in this preferred area will, via the budget constraint, reduce the amount of money available for purchase on non-location commodities and thus reduce the level of utility enjoyed. Given freedom to move in search of better housing opportunities, this type of bidding will raise demand in some locations and lower it in others to the point where all utilities for this group have been equalized. This implies that there is a competitive equilibrium and the assumption for freedom to move is again important in achieving this equilibrium. See household groups *A*, *B*, and *C* of a high income class trading off their preference between housing, accessibility, and amenities expenditures in Figure 2.12(a). This contrasts with two households *B* and *X* in a high and low income class respectively shown in Figure 2.12(b).

Given that the utilities of any particular locating group are fixed at any particular point in time, the v which appears in Equation 2.22 is a constant, and we redefine it as

$$v = \ln I' + \ln F \tag{2.23}$$

Since we are dealing with a homogeneous income group, ln I' is a constant and F is an arbitrary constant whose role will appear below. If we now substitute

Figure 2.12 UTILITY FUNCTION AND BUDGET



SOURCE: Adapted from Yeates and Garner (1980). Reprinted with permission.

$M' = I' - r - T$ and Equation 2.23 in Equation 2.22 and rearrange terms, we arrive at the following expression:

$$\ln([I' - r - T]/I') = \ln F - \alpha_1 \ln H - \alpha_2 \ln A - \alpha_3 \ln C' \qquad (2.24)$$

This is an estimating equation which can be empirically tested and which expresses the proportion of non-location expenditures undertaken by each family as a function of housing, accessibility, and community amenities in each location. This equation has two essential properties. First, all the variables in it can be observed for a number of different household classes in a number of different locations, and consequently it can be determined. The level of non-location expenditures can be estimated from this equation and then, since I' and T are known, the rent which would be offered can be estimated using this equation.

We will show how this important procedure can be achieved. If we exponentiate Equation 2.24, we get $(I - r - T)/I' = FH^{-\alpha_1}A^{-\alpha_2}C'^{-\alpha_3}$. Rearranging terms, we can isolate rent on the left-hand side of the equation. We show this value of rent as an estimated value:

$$r = I' - T - I'FH^{-\alpha_1}A^{-\alpha_2}C'^{-\alpha_3}$$

This is equivalent to the form of the budget equation $r = I' - T - M'$. These values of r are bid-rents discussed by Alonso in his development of the theory of location behavior. Expressing $\ln(1 - [r + T]/I')$ in Equation 2.24 in series, and recognizing that $(r - T)/I'$ is a fraction, an approximation can be made only by taking the first term of the series expansion[4]:

$$\ln\left(1 - \frac{r + T}{I'}\right) \approx -\frac{r + T}{I'} \qquad (2.25)$$

This says that our dependent variable is approximately equal to the (negative) fraction of income spent on rent and transportation combined. This is analogous to the dependent variable of many of the housing market analyses: the rent-income ratio.

Notice the location expenditure is small compared to the rest of the budget for a majority of the population. The fraction of income spent on location expenditures can be estimated by this simple formula; it serves as an approximation for the dependent variable in the Equation 2.24. The above analysis indicates that there is substantial uniformity in the behavior among groups that have been defined on socioeconomic grounds. This behavior can be characterized through utility functions of a fundamentally simple nature. Data are available in the census and elsewhere for providing values for these estimates. All of the relevant variables that we suggested on a priori basis turn out to be statistically significant. The uses to which this analysis can be put must be discussed in conjunction with modeling the market clearing mechanism for housing. (See the Herbert-Stevens model in Chapter 4.)

B. Minimum-Cost Residential Location

Alonso's model of residential location would hold that households are located to minimize the cost of housing and travel. For a monocentric metropolis, this cost is expressed simply as $C(d) = H + r(d) + a'Vd/l'$, where $C(d)$ is the total location cost as a function of distance from the metropolitan area's center, the land area desired for the parcel of land is assumed constant, $r(d)$ is the cost of a unit-of-land as a function of location, a' is the unit cost of commuting (cost per unit-of-distance-traveled), d is the location's distance from the workplace at the metropolitan center, l' is the real discount rate on commuting trips due to such modern day conveniences as telecommuting, and V' is the number of one-way commuting trips taken per year (Lund and Mokhtarian 1994).

Since households are assumed to minimize this cost in their location decisions,

$$\dot{C}(d^*) = \dot{r}(d^*) + a'V/l' = 0 \qquad \text{or} \qquad \dot{r}(d^*) = -a'V/l' \tag{2.26}$$

where the derivatives are evaluated at d^*, the least-cost residential location. Inasmuch as land prices tend to decrease with distance from the metropolitan center, $\dot{r} < 0$. So long as this relationship holds and to the extent that telecommuting lessens the number of work trips per year ($V_1 < V_0$), telecommuting is associated with a more gentle land-rent gradient:

$$\dot{r}(d^*)V_0 < \dot{r}(d^*)V_1 < 0 \tag{2.27}$$

Assuming that land prices follow a conventional exponential decay, then $r(d) = r_0\exp(-Kd)$, where r_0 is the land price at the metropolitan area center and K is a decay constant. Therefore,

$$\dot{r}(d) = -r_0K\exp(-K_0) \tag{2.28}$$

Combining Equations 2.26 and 2.28 yields $r_0K\exp(-Kd^*) = a'V/l'$. This results in the least-cost residential location

$$d^* = (l/K)\ln[l'r_0K/a'] - (\ln V)/K \tag{2.29}$$

Notice that this relationship consists of a constant term that does not vary with commuting trips per year, minus a term that increases logarithmically with the number of annual commuting trips.

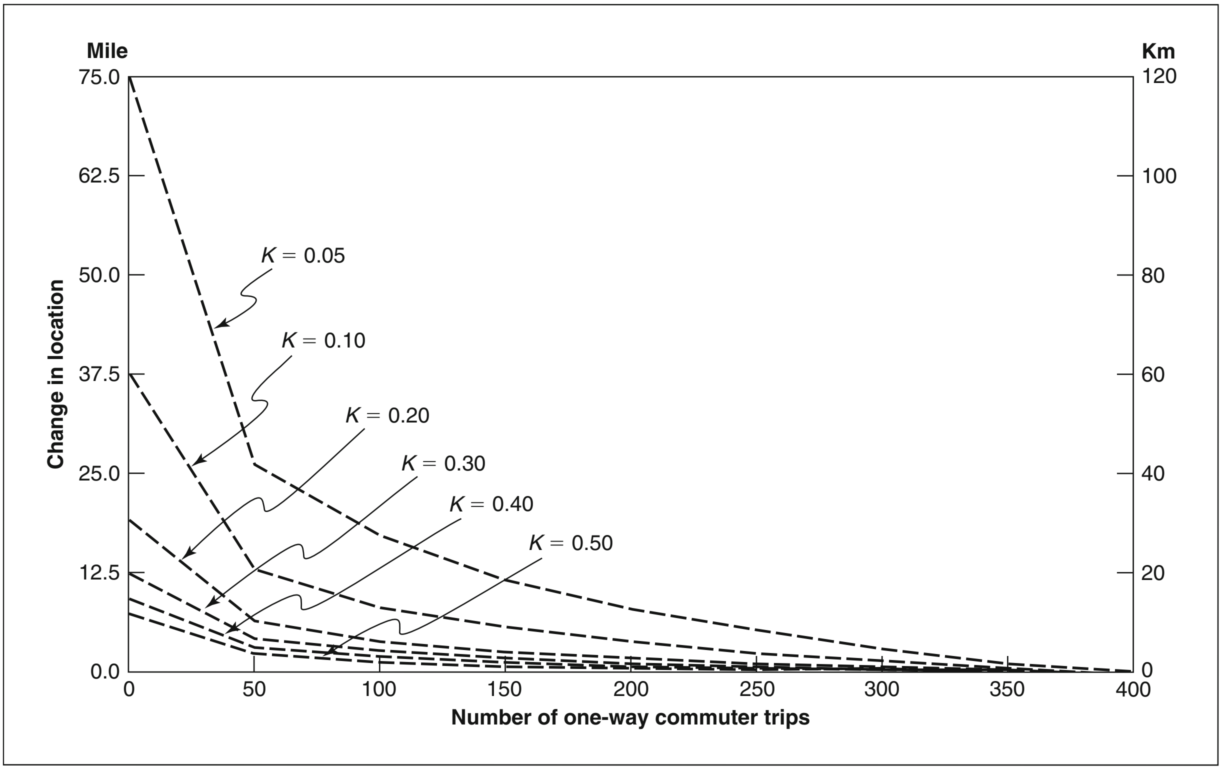
How would residential location change with the onset of telecommuting? To examine this, we define the change in least-cost location,

$$\Delta d^* = d^*(V_1) - d^*(V_0)$$

Replacing Equation 2.29 into this definition yields

$$\Delta d^* = [\ln V_0 - \ln V_1]/K = \ln(V_0/V_1)/K$$

Figure 2.13 CHANGE IN RESIDENTIAL LOCATION WITH TELECOMMUTING



SOURCE: Lund and Mokhtarian (1994). Reprinted with permission.

Note that this change in equilibrium location is affected by only the change in commuting trips and the decay constant of land prices. Other factors entering into the initial location decision do not affect the magnitude of change in the equilibrium least-cost location (Bonsall and Shires 2006).

Example
Consider a household initially located 6.25 miles (10 km) from the metropolitan center ($d_0^* = 6.25$ mi) where 400 one-way commuting trips are made annually ($V_0 = 400$). Land prices decay exponentially at a constant rate ranging from 8 percent to 80 percent per mile (5 percent to 50 percent per km) or $K = 0.08$ to 0.8 per mi. Figure 2.13 shows the change in equilibrium residential location as a function of the number of commuting trips and land prices. It confirms the theoretical and intuitively appealing finding in Equation 2.27, that residential location is affected most by telecommuting in a sprawling city with long commuting distances. ■

V. THE LOCATION DECISION

The above residential location discussions, particularly Equation 2.26, can be carried over to industrial activities.[5] Assume that all activity takes place on a featureless plain consisting of land of equal quality. The rent that any producer will be prepared to pay for a given unit of land i, r^i, will be determined by its output (the number of customer visitations) V, the price per unit at the market, γ, direct cost of production, c, the transport rate per unit of distance a', and d_i, distance from the market:

$$r^i = V(\gamma - c) - Va'd_i \tag{2.30}$$

Here V, γ, c, and a' are assumed constant under conditions of perfect competition. This maximum rent, also referred to as bid-rent by Alonso (1960), is determined uniquely by the location of the site.

A. Bid-Rent Curves

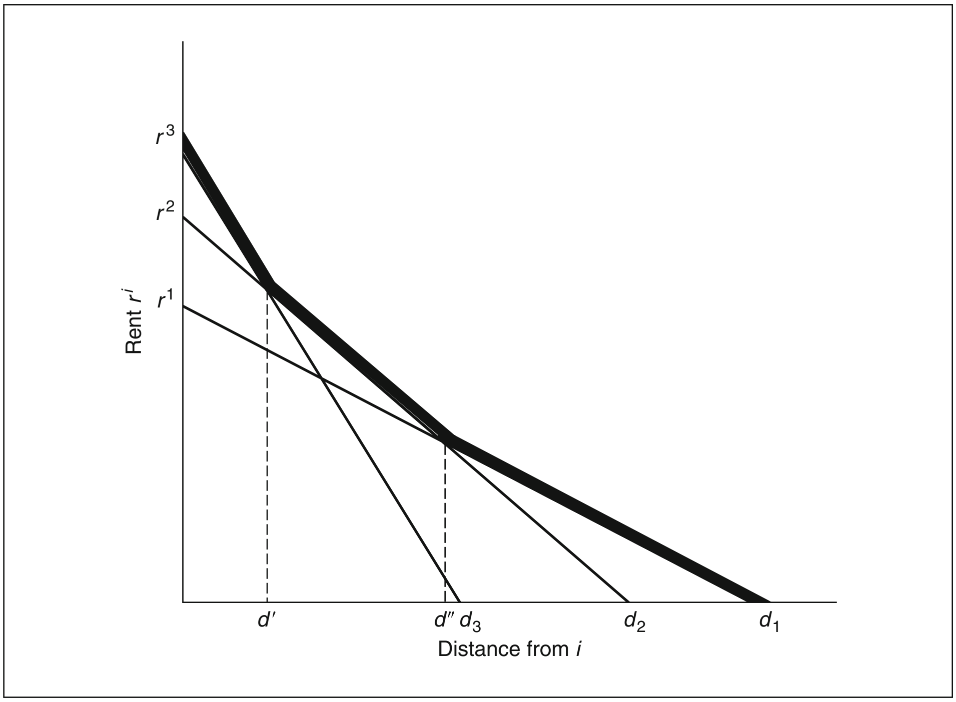
Thus far we have assumed a single activity. If we introduce a second activity, it is obvious that V, γ, and c will not be constant and also it is likely that a' will vary according to weight or any special carriage requirements of the product. However, since perfect competition and freedom of entry prevail, we would not expect the profitability at the most favored location, which we can assume to be arbitrarily close to zero, to differ. The reason is that it and all producers would change production with consequent changes in price to restore an equality of profit. Hence the only change to be made if we have more than one activity is to introduce a', the transport rate, as a determinant of r^i. It is then obvious that by knowing the transport rates for commodities we can derive the location pattern of production about the market. High transport cost activities will locate at a close distance and low transport cost activities will take locations further away. We can determine a relationship between r and d for each a'; the maximum r^i payable at each d_i will determine the activity which will locate there.

Following Alonso (1964), this is best illustrated with a series of bid-rent curves as shown in Figure 2.14. Each bid rent curve $r^i d_i$ is defined by the linear Equation 2.30. Points d' and d'' define important switch points in land use between activities with different bid-rents. The piecewise linear line highlighted in bold is the revealed rent function for the area on the basis that land is allocated to the highest bidder.

B. Industrial Location

Weber (Friedrich 1929) also started with the basic premise that particular locations do not have cost advantages in the actual manufacture of goods. However, in addition to land, most manufacturing industry requires inputs of more than one factor of production and, unlike land, these other factors cannot be assumed to be uniformly distributed in general. The location of a plant will therefore depend on the relative pulls of the various material locations and the market. Weber assumes these to be points rather than areas for simplicity. Assuming that for a particular product these various points are not coincident, the critical factors to be considered will be the relative weights of inputs and outputs and the distances over which these relative weights of input and outputs must be moved. Since transport rates depend on these two factors, the main interest was whether industries would locate nearer the market or to the source of materials and this could be related, through the transport costs, to whether the production process was weight losing or weight gaining. The materials index, the ratio of material

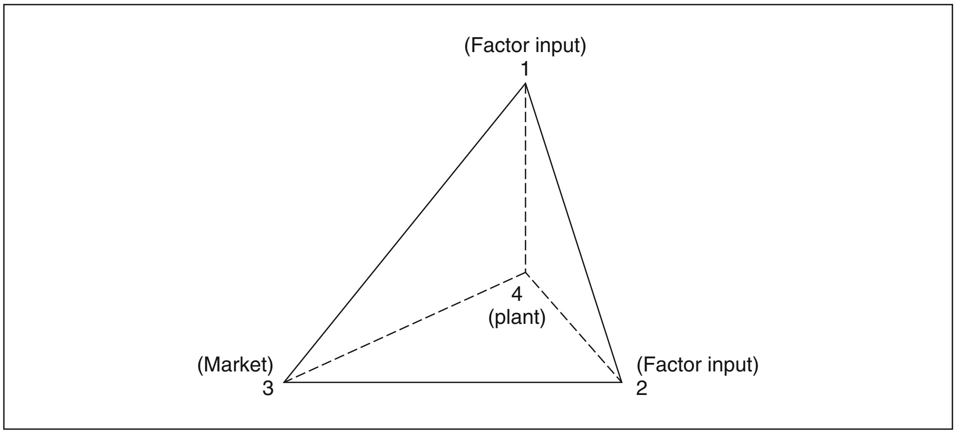
Figure 2.14 BID-RENT CURVES



weight to product weight, is a crude measure. It suggests that high values would involve a location dominated by sources of materials and low values (less than unity) would involve market domination, while values of about one would suggest location indifference.

The basic location criterion is thus minimizing total transport costs, assuming that market price of the product and prices of factor inputs are given and independent of location. The optimal location involves finding a set of

Figure 2.15 WEBER'S INDUSTRIAL LOCATION MODEL



distances d^i the inputs must be moved and distance-to-the-market D: $w_1 d_1 + w_2 d_2 + \cdots + w_n d_n + D$. Here w_1 and w_2, are the inputs required per unit of output. Figure 2.15 illustrates the simplest case of such a model. The figure shows a location triangle relating the market, node 3, to the two factor inputs at 1 and 2. The distances 3-1, 3-2, and 1-2 are geographic distances between the points. The optimal location for a plant at node 4 depends on the effective forces represented by the lines linking it to each corner. These forces are proportional to the relative weights of inputs or outputs as taken into account in the materials index. Node 4 can be found by constructing circles representing isocost lines centered on each corner of the triangle and examining their intersections. The most interesting result from this model is the dominance of end-points, many of which appear optimal, in-between points are of little importance. Numerical examples of this result are shown in Chapter 4.

C. Residential Location Models

According to Alonso (1964), the consumer looking for a housing location maximizes a utility function $v = v(x, s', d)$ where x is the quantity of a composite consumption good representing other activities engaged in by the consumer, s' is the average-size of site, and d is again the distance from the subarea of interest. In his/her location decision, the consumer is constrained by his/her available budget b^U, $p''x + r^i s_i' + a'd_i \leq b^U$, where p'' is the price of the composite consumption good. It is from this model that the bid-rent function for each individual can be derived as the maximum amount a person is willing to pay for a site that would be just as desirable as another.

If we interpret the value of r^i in the above model as being the bid-rent for that location, then from the maximization exercise, we derive

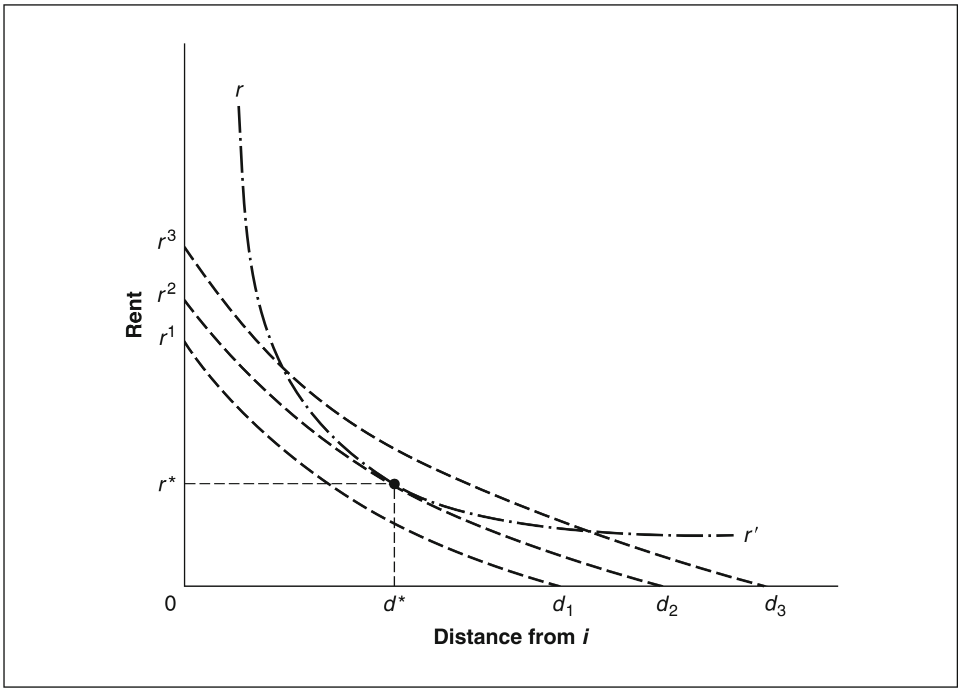
$$\frac{\partial r}{\partial d} = \frac{p''}{s'}\frac{U_d}{U_x} - \frac{1}{s}\frac{\partial(a'd)}{\partial d} \tag{2.31}$$

where U_d and U_x are the appropriate marginal utilities of location and the composite consumption good. Rearranging Equation 2.31 in terms of marginal rates of substitution, we obtain

$$\frac{U_d}{U_x} = \frac{1}{p''}\left[s'\frac{\partial r}{\partial d} + \frac{\partial(a'd)}{\partial d}\right]$$

The above equation states the following: The incremental satisfaction from relocation (in terms of movement outward), which is obtained by substituting travel for goods, must be exactly equal to the cost of that relocation in terms of changing rent costs and changing travel costs. For simplicity we can assume that the good x has a price of unity such that $1/p'' = 1$. Furthermore, since the marginal rate of substitution is assumed to be conventionally negative and since transport costs will increase with distance, the land costs term must be negative. Obviously sites must always have a non-negative size and hence $\partial r/\partial d < 0$; we thus have the basic result that rents must decline with distance and hence the normal assumed shape of the bid-rent curve of Figure 2.16. In this figure, the lines r^i-d_i represent bid-rent curves for an individual household. The higher the

Figure 2.16 HOUSEHOLD LOCATION MODEL



curve, the lower the level of satisfaction. The curve *r*-*r′* is the equilibrium rent function for the city formed as an envelope curve to the various bid-rent lines of Figure 2.14. The equilibrium rent and location for this household is represented by (d^*, r^*).

VI. SCALE AND NUMBER OF PUBLIC FACILITIES

Consider a homogeneous service to be distributed over some spatially distributed population. Let us assume that the service is distributed from a point-representable system of approximately up to four facilities—p_1, p_2, p_3 or p_4—each having an identical scale $\overline{P}$ measured in terms of capacity, capital outlay, or some other metric. The service is consumed by individuals who travel to the facilities for this purpose, and the service is priced at zero, meaning a public service provided by government to the citizens in the area. Total consumption Q of the service is the measure of effectiveness.

A. Static Short-Run Equilibrium

Now total consumption Q is a function of scale P and the number of facilities p

$$Q = Q(\overline{P}, p) \tag{2.32}$$

Total cost of the system C_t is made up of capital cost C_s and operating cost C_o, $C_t = C_s + C_o$, where

$$C_s = C_s\,(\overline{P}, p) \tag{2.33}$$

and

$$C_o = C_o(V) \tag{2.34}$$

In other words, capital cost depends on the number and scale of facilities built, while operating cost is related to the number of consumers served (V). The spatial pattern of facilities for a given $(\overline{P}, p)$ is that pattern for which V is maximized. There exists a fixed budget b^U between capital and operating expenditures.

Figure 2.17, Figure 2.18, and Figure 2.19 illustrate some likely properties of Equations 2.32 through 2.34. Since the service is zero-priced, there is presumably some upper limit V^* to the amount that a population might be expected to consume. Holding the number of facilities constant in Figure 2.17, positive variation in scale may be expected to produce first increasing then decreasing positive variations in demand. The curves in Figure 2.17 actually represent a family of sections through the surface of Equation 2.32. They are therefore demand or consumer coverage curves for the service, given a fixed number, p_k, of facilities at varying scales. Scale expenditures play a role of negative prices or subsidies. An exactly analogous diagram could be made for the number of facilities, holding scale constant. The general character of $V\,(\overline{P}, p)$ is thus a function monotonically increasing to some asymptote V^*. It would look like a curved surface climbing away from the origin.

Figure 2.17 COVERAGE OF CONSUMERS

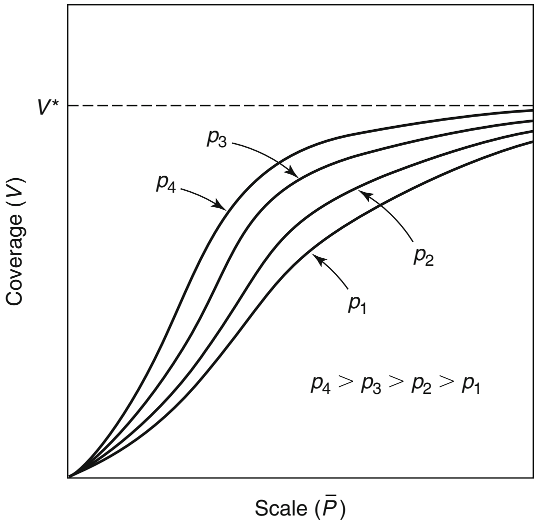
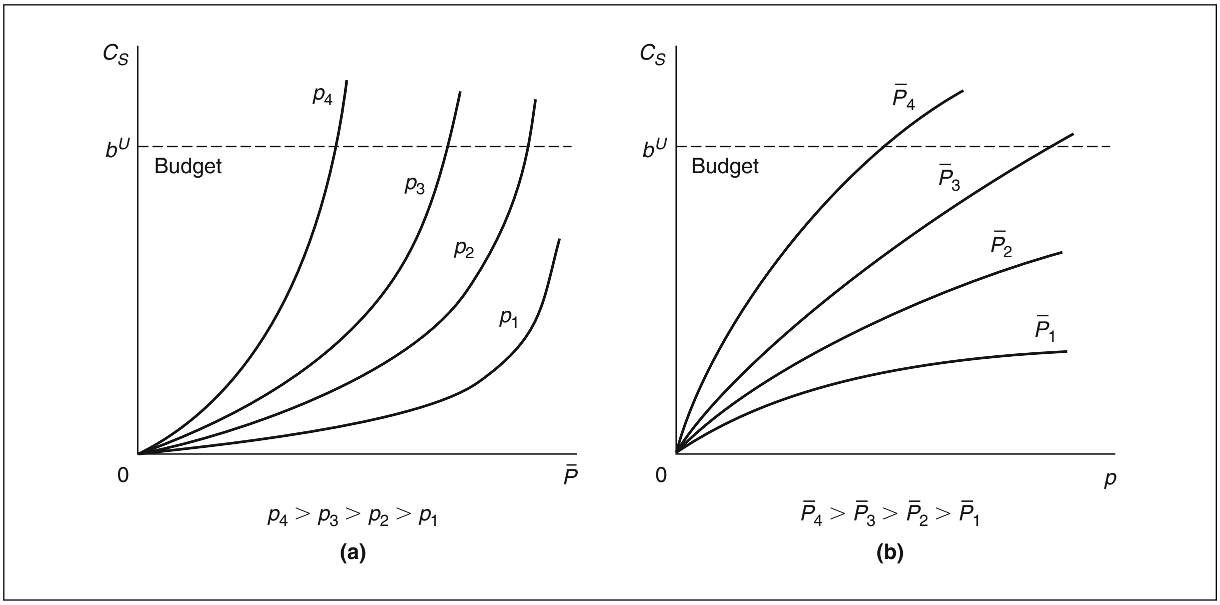
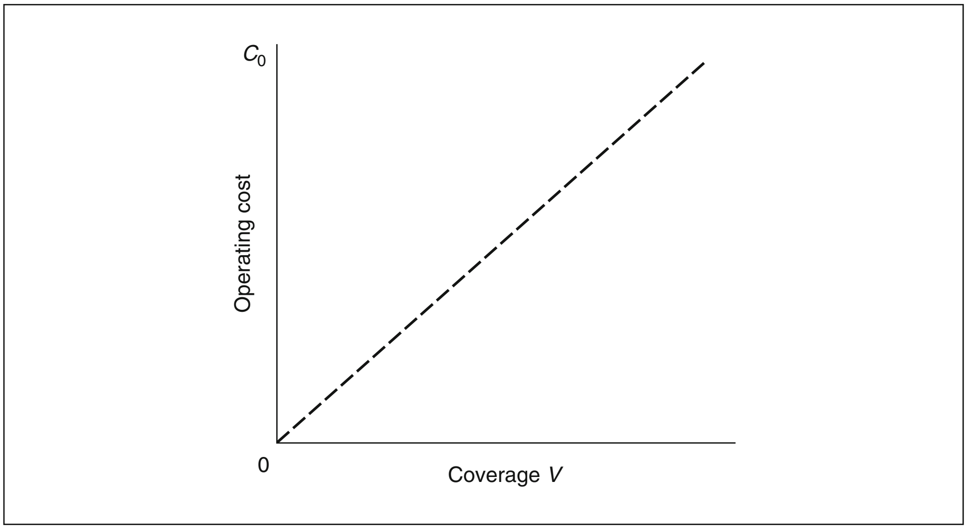


Figure 2.18 CAPITAL COSTS



Cost relationships may be handled in a similar way. Figure 2.18 presents a pattern of capital cost variations for constant levels-of-scale and number of facilities respectively. Although we assume that increase in scale eventually incurs higher marginal cost, there seems to be no reason for such an increase with the replication of facilities. Rather, the reverse seems to hold. The capital cost surface, C_s, may be generated from the families of sections in Figure 2.18. In short, increase in scale results in lower marginal cost compared with construction of new facilities in the beginning, and reverses itself as the system expands to full size.

Figure 2.19 OPERATING COST



For operating cost, C_o, several problems arise. We have made it a function of total demand on the assumption that the marginal product for any variable input to a given system does not vary with the form of the system itself, but only with the aggregate quantity of services demanded and produced. The reason for the distinction between capital and operating costs should be clear. The latter depends upon demand, representing the variable cost of responding to demand at the level induced by the former. In part, this may be an artificial distinction. Demand for a service does respond to the level of variable inputs insofar as it determines convenience and quality of service. We will avoid this complication for the moment by assuming that variable inputs are added to maintain some constant level of quality. For simplicity, this relationship is represented as generally linear in Figure 2.18, although it should be noted that in terms of the variables of Figures 2.17 and 2.18, it is likely to be nonlinear.

With appropriate assumptions about continuity and well-behaved functions, the problem may now be formulated as a constrained maximization: *Max* $V = V(\overline{P}, p)$ subject to $C_t = b^U$. The Lagrangian for this problem is $z = V(\overline{P}, p) - \lambda[C_s(\overline{P}, p) + C_o(V) - b^U]$ for which the conditions for maximization become:

$$\frac{\partial z}{\partial \overline{P}} = \frac{\partial V}{\partial \overline{P}} - \lambda\left[\frac{\partial C_s}{\partial \overline{P}} + \frac{\partial C_o}{\partial V}\frac{\partial V}{\partial \overline{P}}\right] = 0$$

or

$$\frac{\partial V}{\partial \overline{P}} = \left[\lambda \Big/ \left(1 - \lambda\frac{\partial C_o}{\partial V}\right)\right]\frac{\partial C_s}{\partial \overline{P}} \tag{2.35}$$

Similarly

$$\frac{\partial V}{\partial p} = \left[\lambda \Big/ \left(1 - \lambda\frac{\partial C_o}{\partial V}\right)\right]\frac{\partial C_s}{\partial p} \tag{2.36}$$

and

$$\frac{\partial z}{\partial \lambda} = C_s(\overline{P}, p) + C_o(V) = b^U \tag{2.37}$$

Combining Equations 2.35 and 2.36, we obtain the maximization condition

$$\frac{\partial V}{\partial \overline{P}} \Big/ \frac{\partial V}{\partial p} = \frac{\partial C_s}{\partial \overline{P}} \Big/ \frac{\partial C_s}{\partial p} \tag{2.38}$$

The equilibrium condition basically says that the maximal coverage is attained by a combination of scale expansion and new facility construction as justifiable by the marginal costs of the two ways to provide capacity. The consequences of our assumption about variable operating cost show up immediately in Equation 2.38. The equilibrium condition for demand maximization includes only system variables. If this seems peculiar, we might reflect that operating cost appears in Equation 2.37, which says that the cost for service coverage and system capacity expansion is limited by the budget available. Given our assumption that a given increase in demand generates the same operating cost no matter whether it derives from the scale or number of system components, its absence from Equation 2.38 is less surprising. Whether that assumption is tenable is another matter.

More significantly, this formulation evades the problem of location via its cost structure, which is totally dependent upon scale and number of facilities and has no spatial cost components. So far the researchers have been unable to incorporate the location problem into a pure analytic model. In view of the numerous mathematical programming and heuristic approaches to this type of problem, there would seem to be advantages to structuring the total problem as a computer model. In analytical terms, this raises the problem of our assumption of continuity in the variable p. Using a calculus-based model, we cannot simultaneously assume it would be continuous for scale analysis and discrete for a location-effective algorithm. Perhaps an iterative estimation process is the way around this problem, but the theoretical result is less precise. In any case, it seems probable that the location problem for public facility systems must be attacked in tandem with system structure and scale. Several problems still remain. Introduction of variable facility scales in a single system is clearly necessary. As soon as this is done, then questions of hierarchy begin to arise.

The static equilibrium treated above is general in the sense that it deals with simultaneous location and scale of all components of a facility system. The equivalent partial problem might be formulated in several ways. If an increment to a budget for an existing system is given, then we might be interested in determining the optimal addition to the system. This does not necessarily mean that any new components are added. The entire budget increment could be spent on scale changes. If the problem is to achieve a specified incremental gain in some effectiveness measure, the same qualifications would apply. In these circumstances it is not clear how a partial form should be specified. Possibly, it should hold the present facility location structure constant and allow only scale changes and new facility locations. Again, advances in more sophisticated methods than simple calculus are necessary for addressing such problems.

B. Dynamic Long-Run Equilibrium

To analyze systems of facilities with static equilibrium analysis is to ignore a most important characteristic: their changes over time. Facility systems are usually built quite slowly, reacting to changes both in the size of the broader systems they serve and in technology and social preferences. If the broader system is a growing city, then there may be conflict between static and dynamic system optima. This may be especially true if, for whatever reason, decisions early in a system's development can effectively close off options for later forms. A geometric illustration of a dynamic system conflicting with static solutions is offered by a simple model. Consider the circular and generally symmetric city represented in Figure 2.20. At this particular size and for some local service, the optimal number of facilities is one, and it is located at the center, *A*. The city grows symmetrically both in density and at its outer margin until it reaches the size shown in Figure 2.21. At this new level the static-equilibrium solution, taking into account a probable larger budget for the service, calls for two identical facilities, *B*. If they

Figure 2.20 FACILITY EXPANSION IN A CIRCULAR AND SYMMETRIC CITY

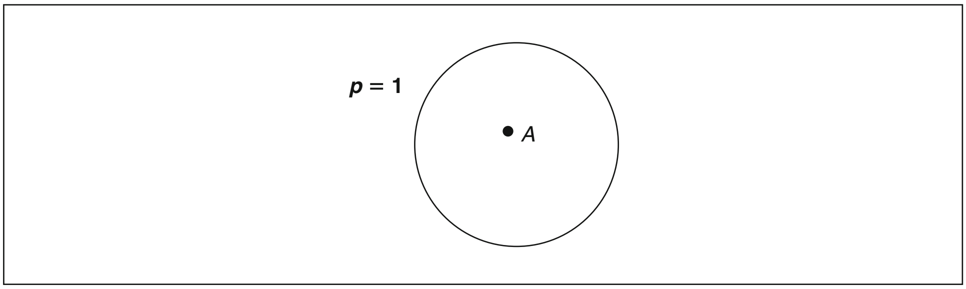
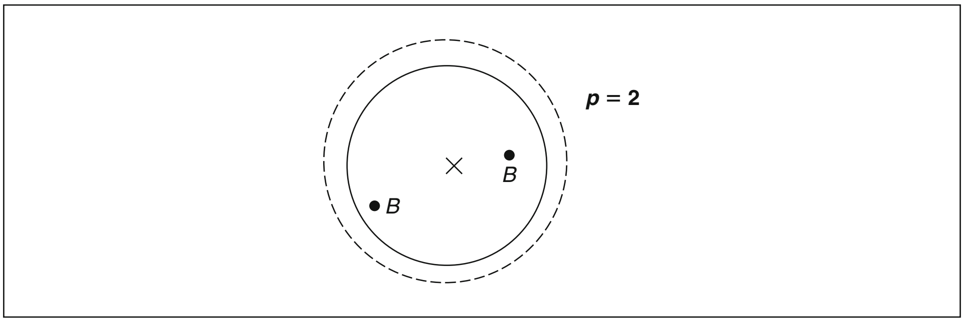


Figure 2.21 LARGER CITY WITH TWO FACILITIES



are located symmetrically, there is no path of growth for this facility system from stage 1 to stage 2 that does not call for removal of *A*. Whether that is likely depends on the rate of growth and the fixed capital investment in *A*.

The example is made artificial by the assertion of identical facilities. In practice, accommodation may be partially achieved by variations in scale among facilities. For example, the equivalent problem for three components might be to approximate a symmetric uniform scale optimum by a variable scale but still symmetric three-component system (see Figures 2.22 and 2.23 respectively). In the latter, the original facility is retained at a larger scale than the others. Without specifying particular forms for the relationships between spatial pattern, scale, and demand, we cannot say much more than this.

The dynamic long-run equilibrium discussion above suggests two modeling approaches. We may look for possible system growth paths through time under varying constraints and criteria for effectiveness and try to identify stages at which such paths coincide with static equilibrium solutions, or we may set up static equilibrium solutions and try to construct minimum cost paths to connect them. Since most facility system analyses are likely to start with an existing set of components, most of which incur high relocation costs, either form could be employed. The choice is perhaps yet another version of the process/end-state conflict in planning models, in this case with both forms involving specific criteria for choice since the decisions are public. Very little work in this direction has been done. Chan (2005) discusses growth paths of land use, rather than facility location, in his chapter on "Bifurcation and Disaggregation." The continuous generalization of facility location—land use—is easier to model inasmuch as it avoids the discreteness or lumpiness

Figure 2.22 THREE FACILITIES AT UNIFORM SCALE

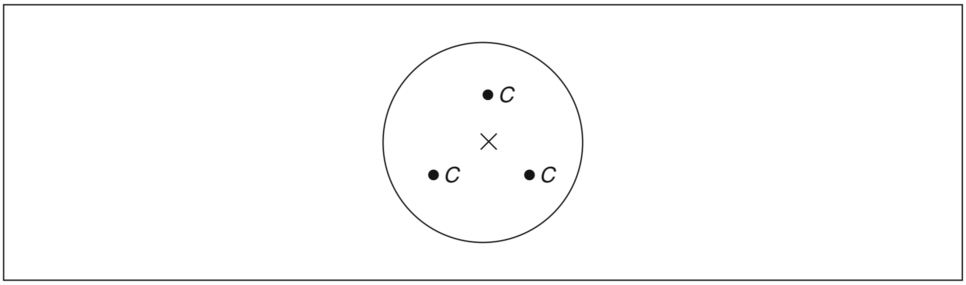
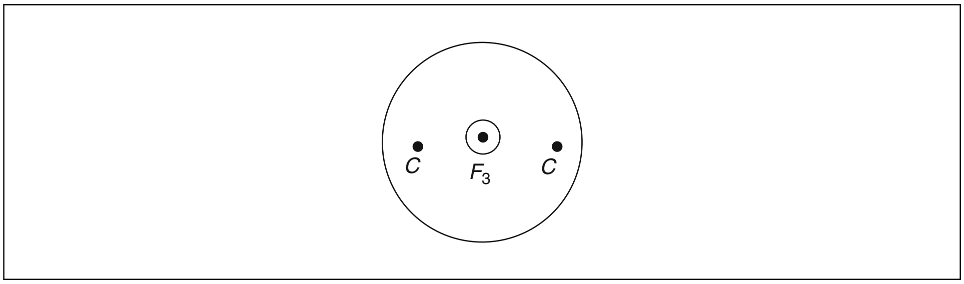


Figure 2.23 THREE FACILITIES AT VARIABLE SCALE



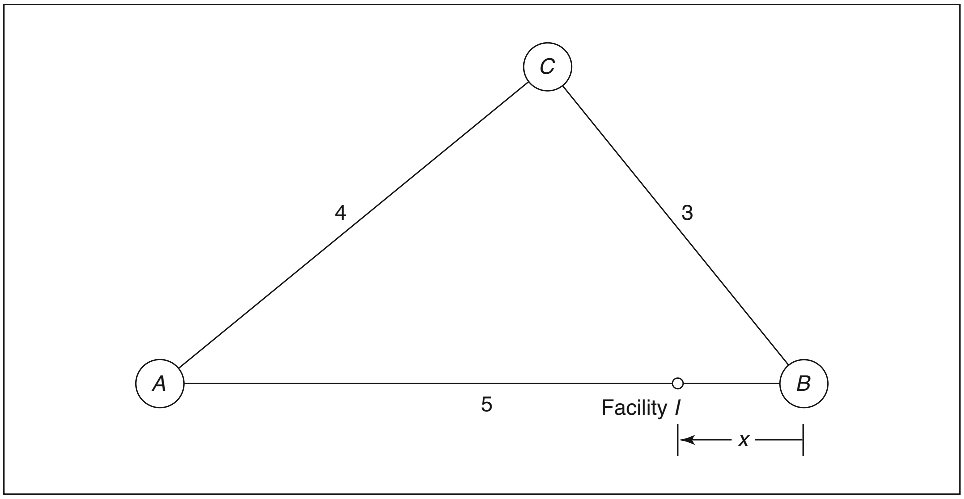
that prevents smooth transition from stage to stage, although bifurcation models do allow for precipitous happenings to take place. Again computer models seem to be most promising given the mathematical complexity of any reasonable looking structure for analysis. (One such program, the Garin-Lowry model, is included on the CD/DVD under the YI-CHAN folder.)

The main problem of locating public services, as can be seen, is choosing the scale and the number of facilities at specified geographic locations that would be most adequate to provide the public services for the budget allocation. The theoretical exposé, while addressing most of the key considerations in planning for public services, has to be further refined for specific applications. Associated with the scale and location considerations, for example, are the ways and means to make the public service available to the community. In this regard, the spatial location of a facility becomes as important as the scale and the number of facilities.

VII. SPATIAL LOCATION OF A FACILITY

Consider the triangular network *ABC* as shown in Figure 2.24, where there are three highways represented by the three edges of the triangle. *A* facility, for instance, a shopping mall, is to be located on the highway system so that the distance to the farthest population center *A*, *B*, or *C* is minimized. The demand at *A*, *B*, or *C* does not enter into the picture in this example; only distances are considered.

Figure 2.24 TRIANGULAR NETWORK *ABC*



A. Center of a Network

Suppose for the time being the facility is to be located among candidate sites on a highway between nodes *A* and *B*, which has a separation of 5 miles (8 km). Let us place a facility at point *I* at a distance of x from node *B*. The distance function between node *A* and point *I* is $5 - x$, and the distance function between node *B* and *I* is simply *x*. These distance functions are shown in Figure 2.25 (Ahituv and Berman 1988). We are supposed to find the one center location, or the location which minimizes the farthest point away. The maximum distance is shown on the upper envelope of Figure 2.25. The minimum occurs at $x = 2.5$ miles (4 km) from *A*, or halfway between *A* and *B*, which is located at the lowest point on the envelope. This is sometimes referred to as the **mini-max** solution.

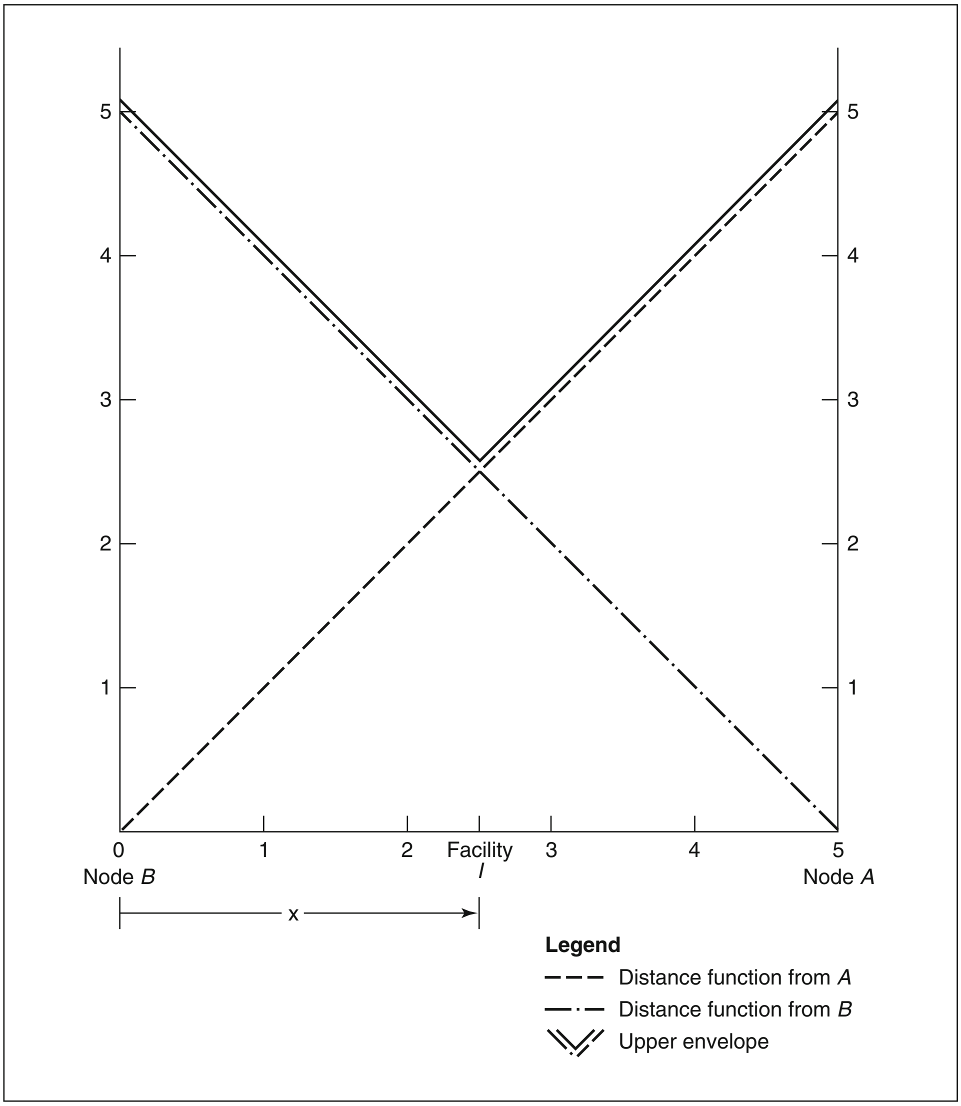
Unfortunately, the problem is more involved, since there is node *C* as well. Let us examine the distance between points on link (*A*, *B*) and node *C*. If the facility is located at node *B*, the shortest distance to node *C* would be 3 miles (4.8 km). When we move point *I* along the link (*A*, *B*) from *B* toward *A*, the shortest distance function becomes $3 + x$. This, however, stops when *x* reaches 3 miles from node *B*, because at that point it is better to approach node *C* via node *A*. The distance function from *I* to *C* becomes $9 - x$, where 9 is the sum of the distances of links (*B*, *A*) and (*A*, *C*), and *x* remains to be the distance of point *I* from node *B*. The complete distance function is given by

$$d_{I3} = \begin{cases} 3 + x & \mathit{for}\ 0 \le x \le 3 \\ 9 - x & \mathit{for}\ 3 \le x \le 5 \end{cases} \tag{2.39}$$

The function is shown in Figure 2.26.

In Figure 2.27, we have combined the distance functions to nodes *A* and *B* from Figure 2.25 with the distance function to node *C* in Figure 2.26. A new upper envelope is drawn, which describes the maximum distance from *I* to nodes *A*, *B*, and *C*, depending on the location of *I* on link (*A*, *B*). The minimum of the maximum distance is obtained when the facility is placed at a distance of $x = 1$ mile (1.6 km) from *B*. At this facility location, the maximum distance to demands at A, *B*, and *C* is minimized at a value of 4 miles (6.4 km). In a similar

Figure 2.25 DISTANCE FUNCTIONS BETWEEN A FACILITY AND DEMANDS AT *A* AND *B*



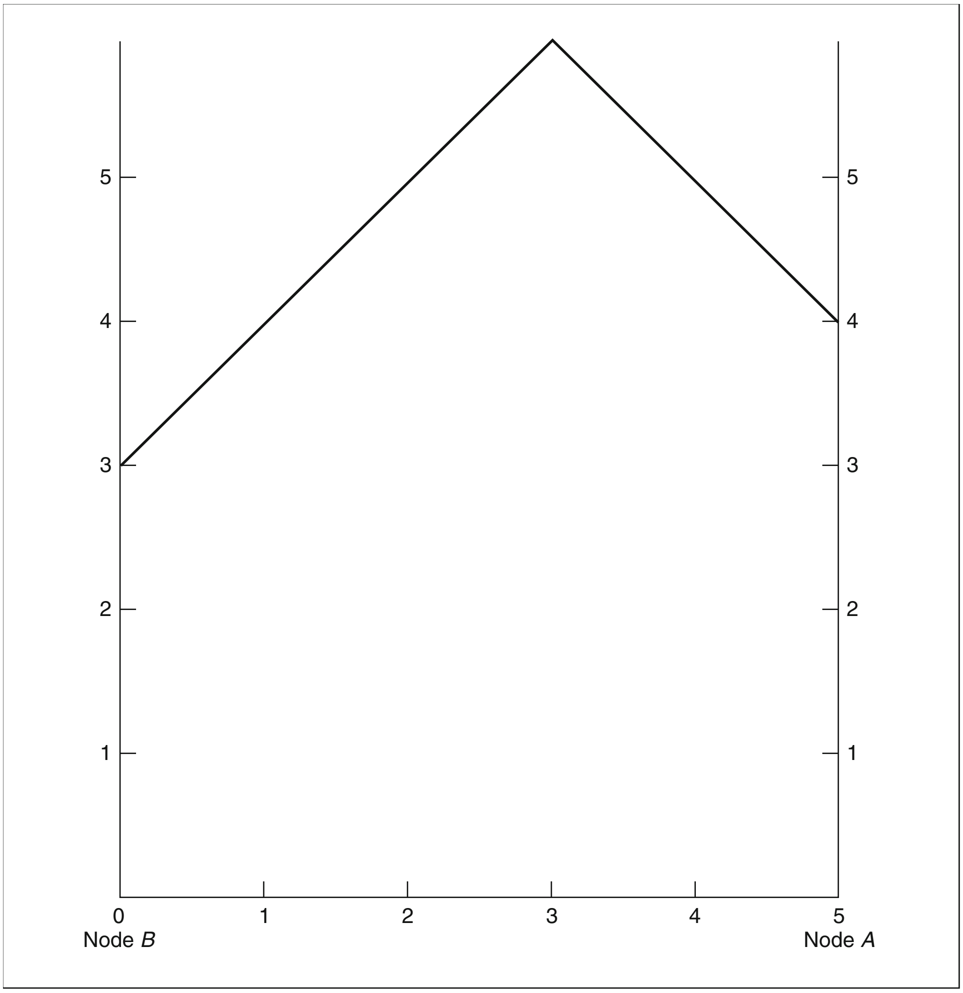
SOURCE: Adapted from Ahituv and Berman (1988). Reprinted with permission.

fashion, we proceed to inquire about the distance functions between points of link (*B*, *C*) and node A, then link (*C*, *A*) and node *B*. The process is in fact quite tedious. More efficient algorithms are available to circumvent this exhaustive search procedure, but they are beyond the scope of this text. Interested readers are referred to the "Facility Location" chapter in Chan (2005).

B. Median of a Network

Suppose we are to locate a facility such that the average distance from a demand node to the nearest facility is minimized—the minimum-of-the-weighted-sum **(mini-sum)** solution. It has been shown (Hakimi 1964) that such a facility has to

Figure 2.26 CENTER DISTANCE FUNCTION FOR LOCATING FACILITY IN A NETWORK

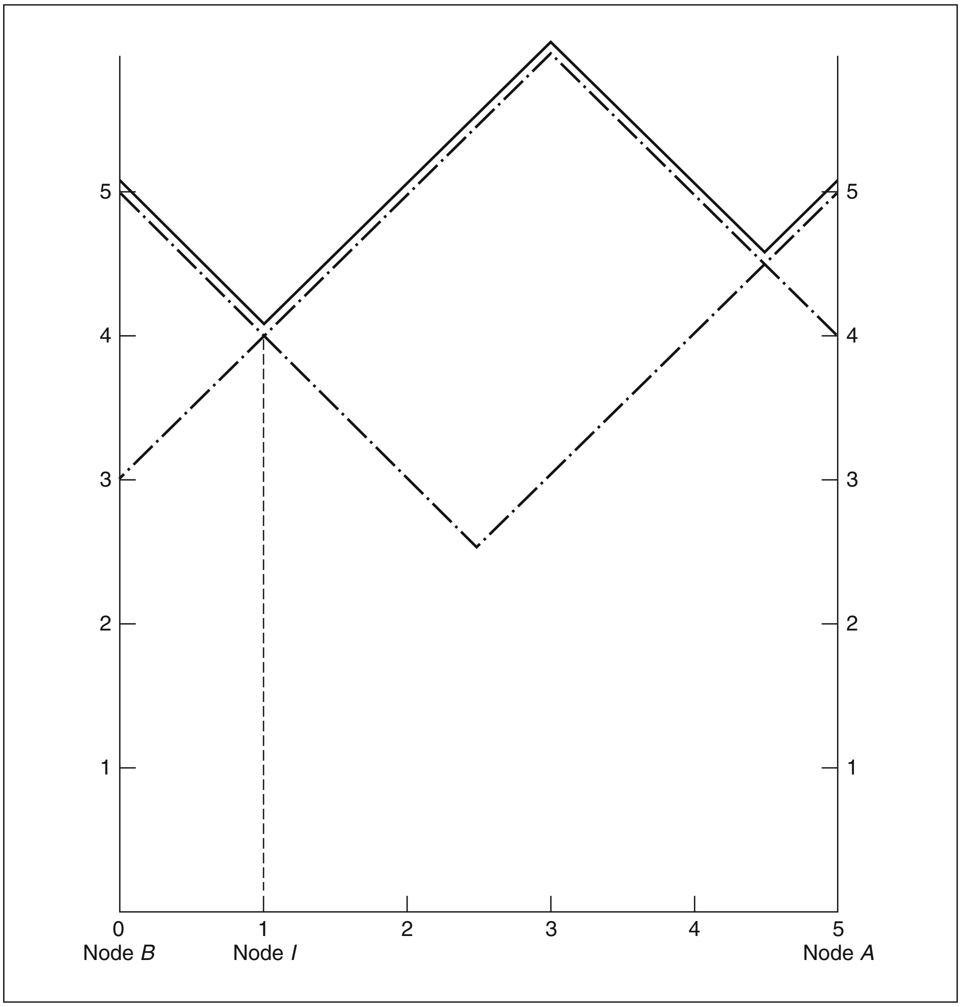


SOURCE: Adapted from Ahituv and Berman (1988). Reprinted with permission.

be located at a node. This is distinctly different from the center problem above, in which the facility can be anywhere on an arc (including the two nodes that define the arc also.) To show this nodal optimality condition for one median, we examine the network consisting of only one link, as depicted in Figure 2.26 (Ahituv and Berman 1988). *A* and *B* represent the two demand nodes, which are separated by a distance d_{AB}. The demand proportion generated at node *A* is T_A', while that at node *B* is $T_B' = 1 - T_A'$. Suppose we place the facility at *I* on link (*A*, *B*). Assume d_A is the distance between node *A* and the facility *I*. The average weighted distance for delivering the service from *I* to the consumers, or for the consumers to access the facility, is

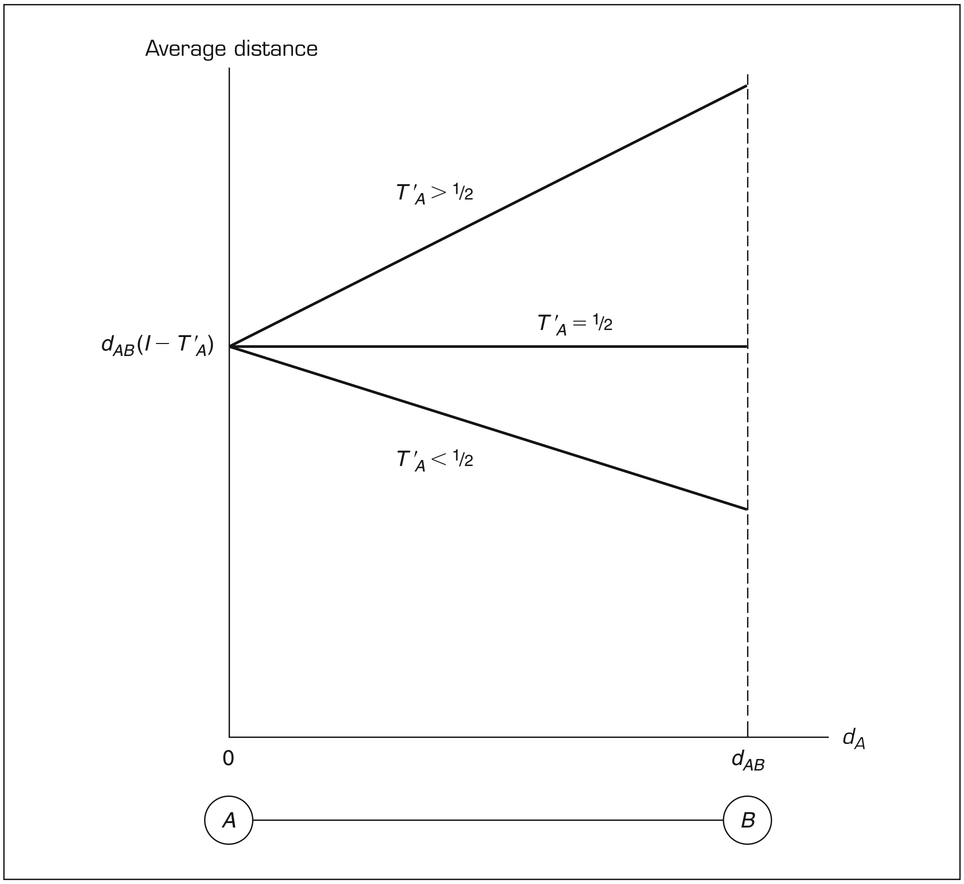
$$T_A'd_A + (1 - T_A')(d_{AB} - d_A) = T_A'd_A - d_{AB} - d_A - T_A'd_{AB} + T_A'd_A = d_{AB}(1 - T_A) + d_A(2T_A' - 1) \qquad (2.40)$$

Figure 2.27 COMBINED DISTANCE FUNCTION FOR FACILITY IN A NETWORK



The first term of the above equation is constant; it does not depend on the location of I. The second term is a function of the location of I, or d_A. Now suppose node A generated more demand than node B, thus $T_A' > {}^1/2$. Hence $(2T_A' - 1) > 0$ and Equation 2.40 is minimized when $d_A = 0$, or when the facility is located at A. However, if node B generated more demand than A, namely $T_A' < {}^1/2$ and $(2T_A' - 1) < 0$, Equation 2.40 is minimized when d_A assumes its biggest possible value d_{AB}. In this case we will place the facility at node B. If the two nodes generate equal demand, facility I may be located anywhere on link (A, B) including the two nodes. Figure 2.28 illustrates the above problem graphically. The average distance as represented by Equation 40 is plotted as a function of the distance from node A to the facility I, d_A. For $T_A' < {}^1/2$, the median should be located at A, where the average distance is minimized. For $T_A' = {}^1/2$, the median can be anywhere between A and B and the travel distance is the same. For $T_A' < {}^1/2$, the facility should be located at B. Figure 2.28 contrasts sharply with Figure 2.25 in that upper envelope in

Figure 2.28 AVERAGE DISTANCE AS A FUNCTION OF MEDIAN LOCATION



SOURCE: Ahituv and Berman (1988). Reprinted with permission.

the latter has a kink in the middle while the former is a monotonically increasing or nondecreasing function. The former identifies a nodal optimum at either *A* or *B*, while the latter locates an optimum in between the two nodes *A* and *B*.

This problem will be discussed again in Chapter 4, where the same problem will be formulated as a linear program, which yields the nodal optimality results directly from the properties of a linear program. From the gravity model, center and median discussions, it is quite clear that depending on the figure of merit for evaluation, a facility can be located at very different places. It is therefore important to properly define an evaluation measure from the beginning of an analysis.

C. Competitive Location and Games

Let us now illustrate competitive location decisions on a network. Suppose there are already *p* facilities located. We wish to locate *r* new facilities that are to compete with the existing facilities for providing service to the customers at the nodes. All demands are perfectly inelastic and the consumers' preferences are binary. We assume customers will change their habits and use the closest new facility if and only if it is closer to them than the closest old facility. Ties are broken in favor of an old facility. Suppose there are two competitors, where both players wish to control as large a share of the market as possible. The first player selects p points for his facilities; the second player, having knowledge of the competitor's decision, selects r points. As the problem is presently stated, each player has exactly one move and has to make the best move possible. This is especially true in situations where the facilities are expensive to construct, and once the facilities are constructed no further moves can be contemplated. The first player knows that once the p sites are selected, the second player will then select the best possible r sites for the facilities. One may pose two possible scenarios for this game to continue beyond the first move by each player (Hakimi 1990).

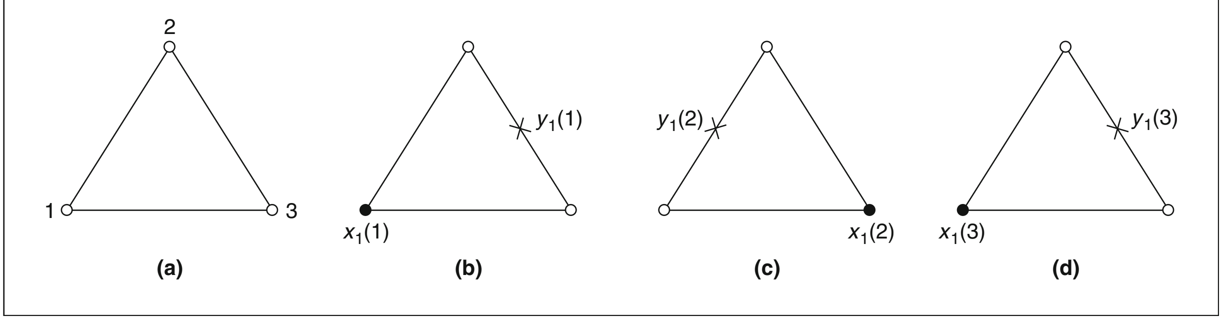
(a) The facilities are mobile but for each player it takes a certain amount of time to respond to the other player's choice of sites (move), assuming that the players do have the computational power to make the best move at each step.

(b) The first player does not have the computational power to find r centers while each player does have the capability of finding r medians or p medians. For both cases, the question arises about where the two players will end up.

Example 1

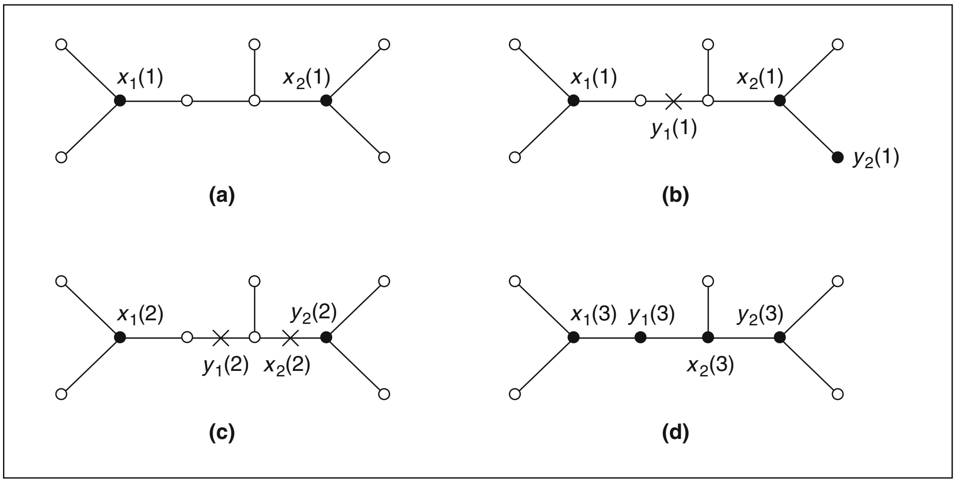
In the example shown in Figure 2.29(a), we assume $p = r = 1$, the payoff at each node to be 1, and the arc lengths are all 1. In Figure 2.29(b) both players' first moves are indicated, where $y_1(1)$ is the mid-point on the edge (2, 3) which is a 1-median. At this stage, it is the first player's turn to move. That move ($x_1(2)$) and the second player's response to it ($y_1(2)$) are shown in Figure 2.29(c). Finally, Figure 2.29(d) indicates the third move of the first player and the second player's response. At this stage, it is clear that the game will continue indefinitely. Whichever player quits first is the loser and will control exactly one-third of the market, leaving the rest to the other player. This example illustrates a situation where the game does not reach an equilibrium, that is, where each player finds that continuing to move is the only way to avoid being limited to the one-third share of the market. Note that in the above example, the first move by the first player, that is the choice of $x_1(1)$, is a 1-center of the network. ■

Figure 2.29 NON-EQUILIBRIUM EXAMPLE



SOURCE: Hakimi (1990). Reprinted with permission.

Figure 2.30 EQUILIBRIUM EXAMPLE



SOURCE: Hakimi (1990). Reprinted with permission.

Example 2

Let us now consider the network of Figure 2.30(a). Assume the payoff at each node is 1, $p = r = 2$, and the arc lengths are all 1. The first player's move $\{x_1(1)$ $x_2(1)\}$ and the second player's response $\{y_1(1), y_2(1)\}$ are shown in Figure 2.30(b). The first player's second move $\{x_1(2), x_2(2)\}$ and, correspondingly, the second player's second move $\{y_1(2), y_2(2)\}$ are shown in Figure 2.30(c). The players' third moves $\{x_1(3), x_2(3)\}$ and $\{y_1(3), y_2(3)\}$ are again shown in Figure 2.30(d). Now it is the first player's turn again. He or she knows, of course, of the positions of his or her competitor and finds that his or her present location is at a 2-median. Thus he or she will not move from his or her present position which implies that the second player also will not move and the game is over. Thus the game terminates in an equilibrium state. We note in passing that the first player's position constitutes a 2-center location of this tree network as well. ■

D. Imperfect Information

It can be seen that the spatial games illustrated above is based in part on the players' lack of perfect information. We start with a single player making a decision on the basis of a known set of information. The first decision to make is whether the player is in the best situation achievable. If he or she is not then he or she must take action. However, there are two problems: one is a lack of perfect information, such that what the player perceives is not necessarily true and because of this ignorance additional information might be needed. Secondly, the player recognizes that even if the adjustment to improve the situation is made, that might not be achieved in a given decision period. In general, our decision maker is assumed to be extremely myopic, to the extent that the system state does not change as a result of his or her decision. We have a situation reminiscent of early attempts to solve classic models of oligopoly markets, in other words, markets dominated by several players. Under these assumptions, the market solution could be shown to be stable, as in the Cournot case where the rival's output is assumed constant in each decision period, and one adjusts his or her output to maximize profit accordingly. Or the market may be unstable, as in the Bertrand or Edgeworth case, where the rival's spatial prices are assumed constant, and the price is adjusted to under cut the rival.[6] In these cases we need to examine two features, whether a full stable equilibrium will be reached and, if so, the speed at which this will take place. The critical factors will be the adjustment to the assumed optimal position and the possible error in making that assumption. Oftentimes, we are concerned less with the final equilibrium itself and more with the path leading to it. In this regard, we are particularly interested in individual players' reactions in each period (Vickerman 1980).

A more realistic model would need to relax the assumed myopia of individuals and introduce strategic reactions of the type adopted in game theory, in which perfect knowledge is assumed. Starting with pure zero-sum games, for example, a conservative player is maximizing minimum gain while the other equally conservative player is minimizing maximum loss. As implied in a zero-sum game, gain to one player matches the amount of loss to the other. In general, individuals are concerned not only with their own attempts to optimize but also with any reactions of conflicting parties to their own actions. A simple example will illustrate the complexities introduced here. A supermarket chain siting a new store will recognize that other shops will be responding to the same stimuli (for example, relative proximity to a new residential area) and that this may generate additional benefits such that the precise site cannot be planned independently. It also realizes that competitors will also respond in an attempt to secure new markets themselves. The calculation depends additionally on the assumptions made about the response of customers, both existing and potential. In the absence of collusion, all of these responses have to be given ahead of time, but the final solution will depend on how good those assumptions are. Once again we shall need to be concerned with whether the path converges ultimately to a stable equilibrium and the speed at which the adjustment takes place. In this case it is not sufficient simply to take assumed responses and examine the behavior of the system, since non-myopic individuals concerned with improving their situations will also learn from revealed responses and accordingly may modify their responses in subsequent decisions. Hence, we also require a learning process within the model.

It will be clear even from this simple description that a representative model of this type will be unavoidably complex. While it would be possible to proceed with continuous functions in a model, there is much to be said for taking a programming approach—an approach which involves systematic computational procedures (often using a computer.) Many of the decisions are of a discrete nature and may involve thresholds and discontinuities that are awkward for a continuous model. The use of discrete time periods also accommodates varying degrees of myopia in adjustment. It is also important that we should stress the operation of the economy as a series of explicitly individual but interdependent decisions. The most useful approach to this type of problem is **recursive programming**, in which a relationship between given system states and expected actions is established, and so are the attempts to simulate a sequence of expected actions through time (Nelson 1971).[7]

There are two possible assumptions about how the markets move into equilibrium at the end of each period. One way is to require the markets to clear period by period, so that a sequence of temporary equilibria is formed, or so that disequilibrium can exist. This was illustrated in Section VI of this chapter, where the transition between one, two and three facilities in a growth environment is anything but continual. An assumption of equilibrium appears unrealistic and almost contrary to the logic of an adaptive model that depends on the independent, albeit linked, reactions of different individuals. Unrealistic as it may be, it does have a number of convenient, simplifying properties. For example, it raises the question of whether individuals attempt to move into full equilibrium. If experience teaches them to modify their behavior, it should also reveal the degree of success of such modification. Given these behavioral adaptations, a policy of suboptimizing may be less costly than an attempt at complete optimization. The sets of reactions might incorporate information about this learning process in a full disequilibrium, wherein it is a conscious decision of individuals that causes the failure to achieve market equilibrium.

It will be apparent that this approach enables a considerable degree of flexibility in the structure and design of a model of the urban, and general spatial, system. At this level of generality it is not possible to draw even qualitative conclusions about whether the results will differ substantially from those of an equilibrium model. It does, however, seem reasonable to expect that, freed from a requirement of a dynamic equilibrium path or even a period by period establishment of equilibrium, the spatial economy may well exhibit a rather different structure. The next step is therefore to use simple versions of this model to simulate the development and structure of urban areas under, for example, different reaction schedules. Such an approach may form an empirical base in the examination of the performance and structure of urban economies under practical planning regimes. A further question is the extent to which such a model can be used to evaluate urban changes, given most evaluation procedures are based on equilibrium metrics. For further details, see chapters starting with "Generation, Competition and Distribution" and ending with "Spatial Equilibrium and Disequilibrium," in Chan (2005). (The reader may also wish to experiment with the software and data contained in the attached CD/DVD under the YI-CHAN folder.)

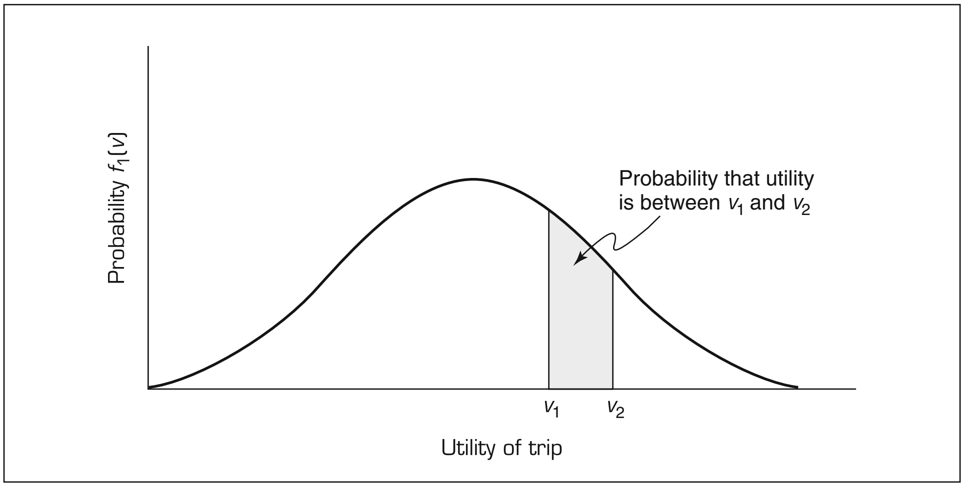
VIII. ECONOMIC BASIS OF THE GRAVITY-BASED SPATIAL ALLOCATION MODEL

In the traditional literature, the most common location technique for land use (as contrasted with facility location) is the gravity model. Here we will derive the various forms of the gravity model based on the assumption that individuals maximize their net benefits in choosing a destination facility (Cochrane 1975). The trade proportions among competing shopping centers, for example, reflect the overall probability of trips being made on the basis of the attractiveness and convenience of the shopping center. Various forms of gravity models have been proposed. They are reviewed below in preparation for later parts of this book.

A. The Singly Constrained Model

Singly constrained gravity model is one in which the number of trips originating in any subarea is assumed determined and fixed. These trips are being made to any of the competing facilities that offer the service. In addition, the model assumes that at each destination there exists some quantity of activities that attracts consumers to patronize that facility. Thus the activity at a shopping mall may be the size of the mall measured in retail floor space. We do not know the

Figure 2.31 PROBABILITY DENSITY FUNCTION OF TRIP UTILITY



SOURCE: Cochrane (1975). Reprinted with permission.

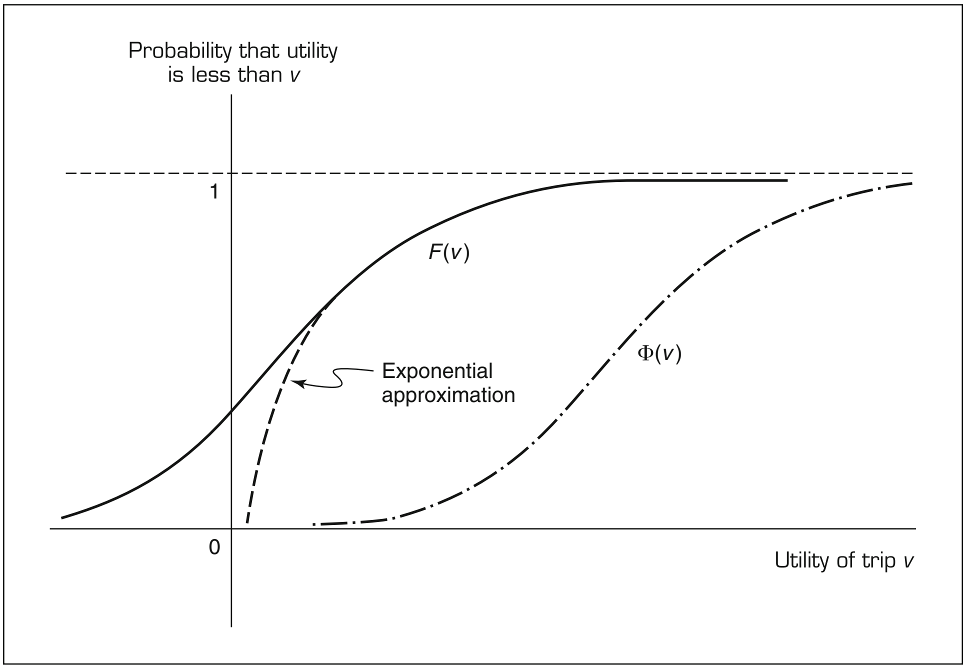
precise value a trip maker might place on any particular trip, since tastes are individual. However, we hypothesize that we can assign a probability that this value will fall between trip utilities v_1 and v_2 (see Figure 2.31.) Define consumers' surplus as the net benefit of any trip after the trip cost has been subtracted from the basic value or utility. Since we can estimate the cost of any particular trip, we can estimate the probability that the surplus lies between any two values.

The central assumption of the present derivation of the gravity model is that the probability that a particular trip maker from one subarea will travel to a facility is the probability that the trip to that facility offers a surplus greater than that of a trip to any other facilities. The probability of an individual trip to a facility being optimal increases with the activity or opportunity at that facility and decreases with travel distance, since the net benefit is reduced by a greater cost. We consider the effect of the number of opportunities offered by a facility. Since we are interested in the probability that the trip to the facility is the best choice, we first estimate the probability of the utility of the optimal (highest utility) trip lying within particular bounds. The cumulative distribution function of the largest v among n independent samples from a common underlying distribution is given by $\Phi(v) = [F(v)]^n$ where $F(v)$ is the cumulative distribution function of the common underlying distribution. The reason is that the cumulative distribution function is the probability that the value is less than or equal to v, and the probability that the best of n is in this range is identical to that of all n being less than or equal to v. Now provided n is moderately large (in double figures at least), $\Phi(v)$ is scarcely affected by the shape of the underlying distribution outside the upper tail (see Figure 2.32.) It is possible to develop an asymptotic (large n) expression of $\Phi(v)$ based only on the shape of the upper tail. If the upper tail can be approximated by a simple exponential function, as indicated in Figure 2.32, $\Phi(v)$ rapidly approaches the simple asymptotic form

$$\Phi(v) = \exp[-ne^{-b(v - \overline{v})}] \tag{2.41}$$

where $\overline{v}$ is the average trip utility.

Figure 2.32 CUMULATIVE DISTRIBUTION FUNCTIONS OF TRIP UTILITY



SOURCE: Cochrane (1975). Reprinted with permission.

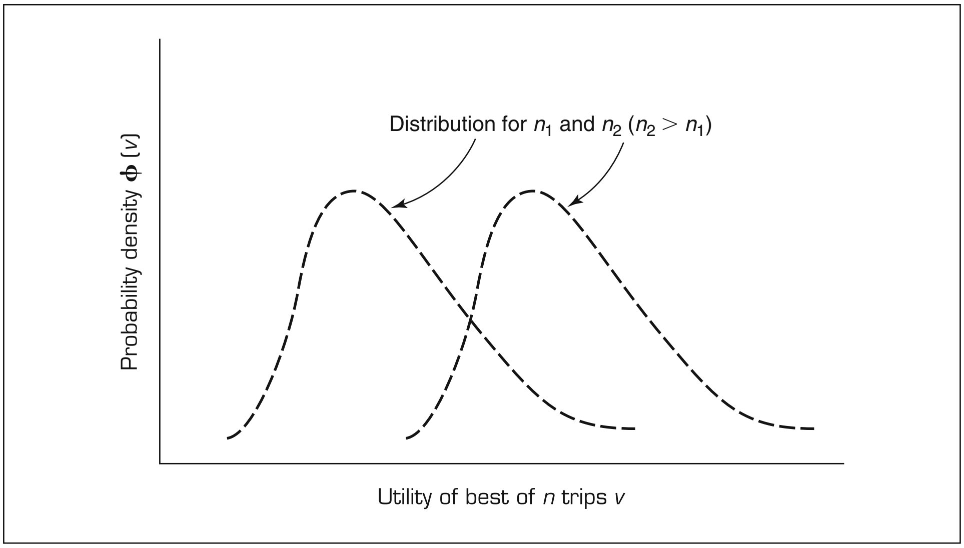
Provided that we assume only that the underlying distribution is approximately exponential in the upper tail, the probability density function for the utility of the best trip in any subarea is given by the differential of Equation 41. This distribution is indicated in Figure 2.33. It is a positively skewed distribution whose skewness is independent of b, $\overline{v}$ and n. The mean is

$$\overline{v} + \frac{1}{b}[\ln(n) + 0.577]$$

and the standard deviation is $\sigma = \pi/\sqrt{6}b$. As n increases, the distribution remains identical in form, but moves to the right of a distance proportional to $\ln(n)$. It may be argued that we do not know the activity at a facility that attracts trips. For our purpose, it is in fact only necessary to assume that the proportion of trips ending up in facility j, $T_j{}'n$, is proportional to W_j, activity at facility j: $T_j'n = c'W_j'$ where c' is a proportionality constant. Hence for trips to facility j, $\Phi(v) = \exp[-c'W_j e^{-b(v - \overline{v})}]$.

We can now calculate the surplus (or net benefit) offered to a trip maker from subarea i by the optimal trip to facility j. We define this surplus as the difference between the probabilistic utility v (the gross benefit of making the trip) and a deterministic trip cost C_{ij} incurred in making the trip. C_{ij} is a generalized cost incorporating direct payments, time costs, and so forth. The surplus is therefore given by $S_{ij}' = v_i - C_{ij}$, and by substitution, we can obtain the probability that the surplus will attain any particular value S′:

Figure 2.33 PROBABILITY-DENSITY FUNCTION FOR THE UTILITY OF THE BEST TRIP



SOURCE: Cochrane (1975). Reprinted with permission.

$$\Phi_{ij}(S') = \exp[-c'W_j e^{-b(s'-\overline{v}+C_{ij})}] \tag{2.42}$$

where $\Phi_{ij}(S')$ is the cumulative distribution function of the surplus accruing from the preferred (optimal) trip between subarea i and facility j. Our basic assumption throughout is that a trip maker will choose the trip from his origin subarea that maximizes personal surplus. The probability that this trip from subarea i will be to facility j is the probability that the highest surplus offered by a trip possibility in facility j is greater than the highest surplus offered by any other facility. This probability is given by

$$\int_{-\infty}^{\infty} \phi'_{ij}(S') \left[\prod_{r\neq j}^{J} \Phi_{ir}(S')\right] dS' \tag{2.43}$$

This equation considers all the joint probabilities that "the surplus resulting from the trip to facility j has a value in the neighborhood of S' ($\Phi_{ij}(S')$) and that "the surplus resulting from a trip to another facility is less than S'." Integrating from $-\infty$ to ∞ assumes that the trip will always be made even if the surplus is negative. However, if the cost determines which trip is made rather than whether a trip is made at all, the probability of the surplus being negative is very low and we can approximate with these limits of integration, which is simpler computationally.

Equation 2.43 can be rewritten as

$$\int_{-\infty}^{\infty} \left[\frac{\phi'_{ij}(S')}{\Phi_{ij}(S')}\left(\prod_{j} \Phi_{ij}(S')\right)\right] dS' \tag{2.44}$$

Differentiating Equation 2.42, we obtain

$$\phi'_{ij}(S') = bc'W_j \exp\left[-b(S' - \overline{v} + C_{ij}) - bW_j e^{-b(-\overline{v} + C_{ij})}\right] \tag{2.45}$$

Substituting Equations 2.42 and 2.45 into Equation 2.44, we obtain

$$\frac{W_j e^{-bC_{ij}}}{\Sigma_j W_j e^{-bC_{ij}}}$$

which is the same as the gravity model of Huff, as indicated in Equation 2.3, except a power function of travel time is now replaced by a negative exponential function of generalized spatial cost. Since the total number of trips originating from subarea i is V_i, the expected number of trips V_{ij} from subarea i to facility j is

$$\frac{V_i W_j e^{-bC_{ij}}}{\Sigma_j W_j e^{-bC_{ij}}} \tag{2.46}$$

which is the customary form of the singly constrained gravity model. We can calculate the total surplus arising from the trips actually made. The calculation uses methods unfamiliar outside statistics (see Cochrane [1975] for derivation):

$$\frac{1}{b}\sum_i V_i \left[0.577 + \ln\left(c' e^{b\overline{v}} \sum_j W_j e^{-bC_{ij}}\right)\right] \tag{2.47}$$

We are normally only interested in the change in surplus resulting from a change in trip costs from C^0 to C', which can be represented by

$$\frac{1}{b}\Sigma_i V_i \ln\left[\frac{\Sigma_j W_j e^{-bC'_{ij}}}{\Sigma_j W_j e^{-bC^0_{ij}}}\right] \tag{2.48}$$

as shown in Equation 2.17 and illustrated in Figure 2.9.

Example

With the appropriate trip-utility, i.e., $W_j = 1$, and a single trip origin $V_i = 1$, Equation 2.46 can be simplified to read $\theta_{ij} = \exp(-bC_{ij})/\sum_j \exp(-bC_{ij})$, Equation 2.47 becomes $S_i' = \frac{1}{b} \ln \sum_i e^{-bC_{ij}}$ and Equation 2.48 becomes

$$\frac{1}{b} \ln \left[\sum_j \exp(-bC'_{ij}) \Big/ \sum_j \exp(-bC^0_{ij})\right]$$

Notice that if the travel-choice set has only one option, the summation sign vanishes and $S_i' = C_i = \overline{v}$. Suppose $b = 0.2$, $C_{i1} = 5$ and $C_{i2} = 8$ for the base-year and $C_{i1} = 5$, $C_{i2} = 8$ and $C_{i3} = 12$ for the forecast year after an accessibility improvement. These three expressions can be evaluated as shown in Table 2.4 and Table 2.5. The second expression S_i'—representing the utility or benefit (actually a disutility

Table 2.4 SAMPLE BENEFIT MEASURES BEFORE ACCESSIBILITY IMPROVEMENT

C_{ij}	b	$\exp(bC_{ij})$	θ_{ij}	$\overline{v}$	S_i'
$C_{i1} = 5$	0.2	0.3679	0.6457	6.0629	2.8126
$C_{i2} = 8$		0.2019	0.3543		
Total		0.5698	1.0000		
$C_{i1} = 5$	0.6	0.0498	0.8581	5.4257	4.7450
$C_{i2} = 8$		0.0082	0.1419		
Total		0.0580	1.0000		

SOURCE: de la Barra (1989). Reprinted with permission.

or dis-benefit in this case) from origin *i*—is evaluated at 2.8126 for the base-year, and 2.0738 for the forecast-year. The third expression, representing the difference in benefit attributable to accessibility improvement, is evaluated at (1/0.2) ln [0.6605/0.5698] = 0.7386 (de la Barra 1989).

If $b = 0.6$, the surplus from origin i is evaluated at $S_i' = 4.7450$ for the base-year and 4.7227 for the forecast-year. The consumers-surplus increase is now 0.0228 (instead of 0.7386.) Remember that the $b = 0.2$ represents a low-sensitivity group while $b = 0.6$ a high-sensitivity group, where sensitivity in this case refers to responsiveness to cost. Thus the lower sensitivity group perceives a lower disutility from the same travel choice set when compared with the high-sensitivity group (2.81 against 4.75 in the base-year). For the consumer-surplus increase, the low-sensitivity group clearly benefits more from the accessibility improvement

Table 2.5 SAMPLE BENEFIT MEASURES AFTER ACCESSIBILITY IMPROVEMENT

C_{ij}	b	$\exp(bC_{ij})$	θ_{ij}	$\overline{v}$	S_i'
$C_{i1} = 5$	0.2	0.3679	0.5570	6.8772	2.0738
$C_{i2} = 8$		0.2019	0.3057		
$C_{i3} = 12$		0.0907	0.1373		
Total		0.6605	1.0000		
$C_{i1} = 5$	0.6	0.0498	0.8472	5.5092	4.7227
$C_{i2} = 8$		0.0082	0.1401		
$C_{i3} = 12$		0.0008	0.0127		
Total		0.0588	1.0000		

SOURCE: de la Barra (1989). Reprinted with permission.

(0.7386 versus. 0.0228). These results show the importance of these surplus indicators in evaluating policy options. Traditionally, transport-related projects have been evaluated with a cost and time criterion, assuming that the preferred project will be the one producing the least of the average-travel-cost $\bar{v}$, where $\bar{v} = \Sigma_{i,j}\, \theta_{ij}\, C_{ij}$. The numerical example above shows that this is clearly a fallacy—$\bar{v}$ has increased from 6.06 to 6.88 and from 5.43 to 5.51, respectively, after accessibility improvement!

Using consumers' surplus, accessibility improvement will always produce benefits, however small, and these benefits will not be the same throughout various population groups. It can be seen, for example, that for the population with a low sensitivity to cost, the percentage of trips destined for the nearest zone, corresponding to $C_{il} = 5$, is 0.6457. By contrast, for the population with a high-sensitivity to cost, the percentage rises to 0.8581. As a result, the average-cost $\bar{v}$ paid by the high-sensitivity group will be lower than that of the low-sensitivity group (5.43 against 6.06). In the forecast-year (after accessibility improvement), 14 percent of the low-sensitivity group can now access the distant zone 3, against only 1 percent of the high-sensitivity group. Correspondingly, the average-cost $\bar{v}$ of the former group rises from 6.06 to 6.88, while the latter group only moves from 5.42 to 5.51. The average utility indicators S_i' show in both cases an improvement when the new accessibility option is introduced, but they also show that the low-sensitivity group benefits more, because the dis-utility moves from 2.81 to 2.07 while the high-sensitivity group hardly moves from 4.75 to 4.72. Hopefully, this numerical example drives home the usefulness of interpreting the gravity model in terms of economic benefits.

B. The Doubly Constrained Model

Aside from a fixed number of trips originating from i, the doubly constrained gravity model also restricts the number of trips ending in j. This model is appropriate for work trips where the number of trips emanating from the origin residential subarea every morning is perfectly inelastic, and these trips are heading toward employment centers that have a specific number of jobs, at least in the short run. If there is no constraint on trip ends, there will be some employment centers j in which the number of unconstrained trip ends will exceed the number of jobs available. We assume that under these conditions competition will lead to the jobs being taken up by those trips for which the surplus available is greatest. This will occur either because the utilities of the set of trip ends are bid down or because the costs are bid up. In either case we may represent the effect as the addition of an extra cost r_j to the trip, these additional costs are set such as to restrict demand to the jobs available.[8]

We then rewrite Equation 2.42 as

$$\Phi_{ij}(S') = \exp\left[-c' W_j\, e^{-b(S' - \bar{v} + C_{ij} + r_j)}\right].$$

Substituting in Equation 2.44 and integrating as before, we obtain the probability of a trip ending up in employment center j:

$$\frac{W_j\, e^{-b(r_j + C_{ij})}}{\Sigma_j W_j\, e^{-b(r_j + C_{ij})}} = \frac{W_j\, e^{-br_j} e^{-bC_{ij}}}{\Sigma_j W_j\, e^{-br_j} e^{-bC_{ij}}} \qquad (2.49)$$

The number of trips from *i* to *j* is correspondingly

$$V_{ij} = V_i \frac{W_j\, e^{-br_j} e^{-bC_{ij}}}{\Sigma_j W_j^{-br_j} e^{-bC_{ij}}}$$

where r_j is the calibration constant chosen such that $\Sigma_i V_{ij} = V_j = c' W_j$ for all *j* as mentioned. It is clear that this model is equivalent to the conventional doubly constrained model

$$V_{ij} = V_i \frac{W_j\, a_{j0} e^{-bC_{ij}}}{\Sigma_j W_j\, a_{j0} e^{-bC_{ij}}}$$

where $a_{j0} = e^{-br_j}$ with both *a* and *r* representing a calibration constant. A numerical example of the doubly constrained gravity model is found in Chapter 3. The change in surplus resulting from a change in trip costs is given by

$$\frac{1}{b} \Sigma_i\, V_i \ln\left[\frac{\Sigma_j W_j\, e^{-br_j} e^{-bC'_{ij}}}{\Sigma_j W_j\, e^{-br_j} e^{-bC^0_{ij}}}\right]. \tag{2.50}$$

In order to balance the number of trip destinations with the number of origins over the entire area, some of the additional facility costs r_j will be positive and some will be negative. These values will result in a_{j0}*'s* less than and greater than one respectively. It should also be noted that the surplus expression represents the benefit received solely by trip makers.

C. The Unconstrained Model

The unconstrained model is the most difficult of the gravity models discussed so far, where the trip generation at origin is modeled in addition to trip distribution. A partially constrained model is suggested by Cochrane (1975) in which it is assumed that there exists an upper limit to the number of trips generated by any subarea—as the trip costs rise, some of the trips are no longer made. When integrating Equation 2.43 above, we took the limits of integration from $-\infty$ to ∞. The low value was used because when the distribution of maximal surplus is very much greater than zero the probability of a negative value of surplus is negligible and we can obtain a simple integral by using these limits. This assumption implies that the primary economic force bringing about trip making is stronger than those that decide the choice between destinations. If this is not the case, we should integrate more precisely between limits of 0 and ∞. This implies that the trip maker decides not to make even the optimal trip if the surplus is not positive. Where the utility of the trip is only of the same order as the cost, this is an important consideration. Certain social and recreational trips are likely to come into this category, although trips such as work trips do not. More will be said about this in the "Location-Allocation" chapter of Chan (2005).

Integrating Equation 2.43 between the new limits leads to

$$[1 - \exp(-b' \Sigma_j W_i\, e^{-bC_{ij}})] \frac{W_j e^{-bC_{ij}}}{\Sigma_j W_j e^{-bC_{ij}}} \tag{2.51}$$

where $b' = -c'e^{b\bar{v}}$. Trips executed V_{ij} can be expressed in terms of this unconstrained model by

$$V_{ij} = V_i(W_i, W_j, b', b, C_{ij})\Theta(W_j, b, C_{ij}) \tag{2.52}$$

where V_i is the trip-generation term and Θ is the trip distribution term. Each of these two terms can be equivalenced to Equation 2.51 by setting

$$V_i = W_i[1-\exp(-b'\Sigma_j W_j e^{-bC_{ij}})]$$

and

$$\Theta_{ij} = \frac{W_j e^{-bC_{ij}}}{\Sigma_j W_j e^{-bC_{ij}}}$$

The trip-generation term constrains the total trips made in response to increases in the cost of trip making. Hence, if costs rise on particular links, the total number of trips changes in accordance with the trip-generation term to a certain limit, and the allocation of trips among destinations changes in accordance with the gravity trip-distribution term meanwhile. Again, we will further develop this model in the "Location-Allocation" chapter of Chan (2005).

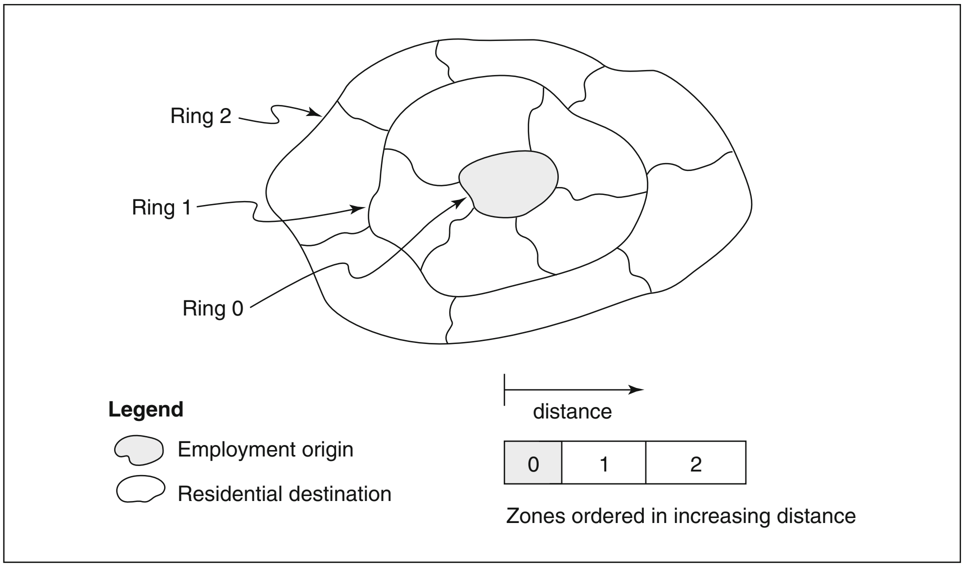
D. The Intervening Opportunity Model

Besides the gravity model, another common spatial allocation model is the intervening opportunity model (IOM). The IOM is based on a probabilistic formulation, which states that the probability, dP, that a trip will terminate in a destination is the joint probability that no termination point has been found among the total number of opportunities n visited so far and that the trip ends up in the current destination which offers an additional dn number of opportunities: $dP = [1-P(n)]\, L'\, dn$. Here $P(n)$ is the probability that a termination point is found in the volume of destinations n, and L' is a constant probability that the subarea visited is in fact the termination point for the trip. Solving the differential equation for $P(n)$, the probability of finding a termination point in the n subareas visited is $P(n) = 1-e^{-Ln}$. The expected number of trips from i, V_i, that will terminate in j, V$_{ij}$, is obtained by multiplying the total number of trips originating at i by the probability that the trip will terminate amid the n_j additional opportunities found in subar*ea j* $V_{ij} = V_i\,[P(n+n_j)-P(n)]$. Substituting the value of $P(n)$ in the above equation, the usual form of the IOM is

$$V_{ij} = V_i\,[e^{-L'n}-e^{-L'(n+n_j)}] \tag{2.53}$$

The basic theory of IOM states that (a) all opportunities are ordered by increasing distance from the origin and (b) the probability of an activity to be located at a particular destination is equivalent to a series of Bernoulli trials, where an activity is more likely to be located closer by than further away, everything else being equal. Thus in the residential location example in Figure 2.34,

Figure 2.34 DEFINITION OF OPPORTUNITIES IN THE INTERVENING OPPORTUNITY MODEL



- □ the probability of locating in destination 0 = L'
- □ the probability of locating in destination 1 but not in destination 0 = $L'(1-L')$; and
- □ the probability of neither locating in destinations 0 nor 1 but locating in destination 2 = $L'(1-L')(1-L')$.

In this example, there are five residential zones at a certain distance away from the employment zone, and there are seven zones yet further away. Here the number of zones within annular ring 0, 1, and 2 are $n_0 = 1$, $n_1 = 5$ and $n_2 = 7$. The zones are identified only by the annular ring in which they are located and all zones are assumed to be of equal size to denote that each offers the same residential opportunities. Alternatively, one can think of the destinations being ordered in increasing distance from the employment origin, each with 1, 5, and 7 opportunities respectively as shown in the lower part of the figure. If the probability of residential location in a zone, L', is $^1/_2$, we can compute the relative frequency of residential activity distribution as

- □ percentage of population living in origin 0 = $e^0 - e^{-(1/2)(1)} = 0.390$
- □ percentage of population living in destination 1 = $e^{-(1/2)(1)} - e^{-(1/2)(6)} = 0.556$
- □ percentage of population living in destination 2 = $e^{-(1/2)(6)} - e^{-(1/2)(13)} = 0.048$

and so on.

The simple numerical example illustrates not only the computational mechanics of Equation 2.53, but also the problem of calibration. For example, we observe that assigning the value of $^1/_2$ to L' is merely arbitrary; its value needs to be calibrated from available trip-length-frequency data. Second, defining residential opportunity as the physical land area may be convenient, but a more workable definition is likely to be problem specific and requires more effort. Finally, it is noted that the population allocation percentages up to the second annular rings do not add up to 100 percent. But if one considers additional annular rings ad infinitum, the sum of the percentages has to be unity according to Equation 2.53. Some practitioners prefer this model on the grounds that it can be developed from a defined set of statistical assumptions. Others have been concerned by the fact that the IOM has no intrinsic cost elements, and in particular does not distinguish the case where the subsequent opportunity is marginally more distant.

Curiously, it is possible to derive the IOM as a special case of the gravity model. We derive these models by assuming a relationship between the cost of transport between two points and the number of intervening opportunities. If we assume this to be of a power form:

$$n = b''[C_{ij}]^{\beta} \tag{2.54}$$

then $C_{ij} = [b'']^{-1/\beta}\, n^{1/\beta}$ where travel cost is not a function of distance as alluded to previously. In the singly constrained gravity model, we can write

$$V_{ij} = V_i \frac{n_j \exp(-b[b'']^{-1/\beta}\, n^{1/\beta})}{\Sigma_j n_j \exp(-b[b'']^{-1/\beta}\, n^{1/\beta})} \tag{2.55}$$

Substituting $b[b'']^{1/\beta} = b_0$

$$V_{ij} = V_i \frac{n_j \exp(-b_0 n^{1/\beta})}{\Sigma_j n_j \exp(-b_0 n^{1/\beta})}$$

and using the incomplete gamma function $\Gamma'[x, y]$, Cochrane (1975) evaluated Equation 2.55 as

$$V_{ij} = V_i \left\{ \frac{\Gamma'[\beta, b_0(n + n_j)1/\beta] - \Gamma'[\beta, b_0 n^{1/\beta}]}{\Gamma'[\beta, b_0 n'^{1/\beta}]} \right\} \tag{2.56}$$

where n' is the total number of opportunities in the study area. The gamma function can be considered as a set of related functions of the second variable, the particular function to be used being indicated by the first variable, which in this case is β. If the number of opportunities is directly proportional to the cost as indicated in Equation 2.54, β is equal to one and the incomplete gamma function becomes the negative exponential function. We then obtain

$$V_{ij} = V_i \left[\frac{e^{-b_0 n} - e^{-b_0(n + n_j)}}{1 - e^{-b_0 n'}} \right]$$

If n' is large, the usual form of IOM results: $V_{ij} = V_i\, [e^{-b_0 n} - e^{-b_0(n + n_j)}]$. This derivation illustrates a very important concept in the analysis of spatial-temporal information. Through spatial cost transformation, apparently unrelated models can be equivalenced. We will have many other examples later on in this book and in Chan (2005) to illustrate this point.

IX. CONCLUDING REMARKS

In this chapter we have reviewed many of the basic economic concepts of facility location and activity allocation. We saw that the determination of spatial patterns—both in discrete facility locations and continuous land-use developments—can be explained in a set of common terms. These common constructs range from median to center models, from input-output analysis to the gravity model—all developed from basic economic concepts such as utility theory. Modern-day econometrics also allows empirically based approaches to be used to forecast future activity patterns. This is performed independent of the classic economic concepts, as illustrated in the interregional demographic projection section. In a fairly readable manner, it illustrates the basic building blocks of spatial-temporal information. To be sure, analysis of spatial-temporal information involves not only economic or econometric techniques, but the well-established economic concepts are convenient and familiar points of departure for many who work in this field.

In the next few chapters, we will provide the ways and means to further operationalize some of these concepts. In Chapter 3, we lay out the statistical procedures; while in Chapter 4, we outline the optimization algorithms. These techniques help to implement what were up to now theoretical constructs in terms of solid operational procedures. Recent advances in both descriptive and prescriptive tools allow us to realize some of the goals that our predecessors can only dream of. We then introduce a more recent paradigm for location decisions, multi-criteria decision making, which departs from traditional economics in several ways. First, it is behaviorally based rather than structurally based, complete with its own version of multi-attribute utility theory. Second, it broadens our concepts of ranking locations and shows that some counterintuitive results regarding transitivity and intransitivity among candidate sites may occur. For example, we demonstrate that site *A* preferred to site *B*, and site *B* preferred to site *C* does not necessarily mean site *A* is preferred to site *C*. Such recent advances in behavioral and mathematical sciences allow for a more innovative approach to modeling spatial decisions in general. It is one of our objectives to report these exciting developments here in this volume.

X. EXERCISES

Self-Instructional Module: PROBABILITY

(to be found on the attached CD/DVD)[9]

An understanding of probability is important to the decision maker. Many decisions must be based on predictions of future events. Inevitably, the prediction of future events has uncertainties and probable errors. An example is population projection, as discussed in Chapter 2 of this text. An understanding of probability concepts helps the decision maker to appreciate the significance of such uncertainties and probable errors.

This activity module is divided into three sections. The *first* section covers some of the theories of probability. The *second* section covers some rules of counting. Finally, the *third* section builds upon the first and second sections and illustrates with some interesting examples.

By the end of this exercise, the student

(a) would be familiar with these concepts: sample space, events, union and intersection of events, empirical or frequency probability, subjective probability, and permutation.
(b) would have seen some useful application of these concepts.

This module serves to introduce or review the fundamental probability concepts, which allows an understanding of what is *information* and *imperfect information*. An example of imperfect information is given in Chapter 2, in connection with locational competitions. Naturally, probability is required for the discussions in Chapter 3, where a number of descriptive analysis tools, including simulation, subjective probability, curve fitting, and information theory are formally discussed.

Probability is a prerequisite for an understanding of statistics, a basic building block of analytics. As such, it serves as an excellent introduction to a subsequent self-instructional module on Probability Distribution and Queuing. It is also a prerequisite for the Appendices entitled "Review of Statistical Tools" and "Review of Markovian Processes".

Problem 1: Gravity Model

The most common theory to explain spatial interaction is the Gravity Model, which states in mathematical terms the relationships between "activity at zone *j*" and "activity at origin zone *i*," as governed by the "spatial cost between them:"

□ Employment at zone $i = E_i$
□ Spatial cost between i and $j = d_{ij}$
□ Proportion of activities from origin i that end up in destination $j = \Theta_{ij}$

We can then write

$$E_i\,\Theta_{ij}\,(d_{ij}) = E_i\frac{N_j/d_{ij}^2}{\sum_k N_k/d_{ik}^2}$$

Correspondingly, populations N_j are to be distributed among the urban area according to

$$N_j = \sum_i E_i\,\Theta_{ij}\,(d_{ij})$$

which, when written for zone 2 in a city of three zones assumes the form

$$N_2 = E_1\,\Theta_{12}(d_{12}) + E_2\,\Theta_{22}(\mathrm{d}_{22}) + E_3\,\Theta_{32}(d_{32})$$

Here,

$$\Theta_{12}(d_{12}) = \frac{N_2/d_{12}^2}{N_1/d_{11}^2 + N_2/d_{12}^2 + N_3/d_{13}^2}$$

and so on.

For the following study area:

From/To	Zone 1	Zone 2	Zone 3
Zone 1	2	8	6
Zone 2	8	3	4
Zone 3	6	4	3
Base-yr pop: N_j	480	870	1020

$$\Theta_{12}(d_{12}) = \frac{870/8^2}{480/2^2 + 870/8^2 + 1020/6^2} = 8.3950 \text{ x } 10^{-2}$$

(**a**) Please calculate $\Theta_{11}(d_{11})$ and $\Theta_{13}(d_{13})$ and verify that the sum of $\Theta_{11}(d_{11})$, $\Theta_{12}(d_{12})$ and $\Theta_{13}(d_{13})$ adds up to the numerical value of unity.

Let us say the total employment in zone 1 is projected to be 500. Accordingly, it means that $E_1\Theta_{12}(d_{12}) = 500 \text{ x } 8.3950 \text{ x } 10^{-2} = 41.975$, or 42 workers from zone 1 will live in zone 2. In order to forecast the *total* number of workers living in zone 2, however, two additional number are needed. They are $E_2\Theta_{22}(d_{22})$ and $E_3\Theta_{32}(d_{32})$. The sum of the three numbers is the total number of employees living in zone 2, in accordance with the equation $N_j = \sum_i E_i\ \Theta_{ij}(d_{ij})$.

(**b**) Please calculate the number of employees living in zones 1 and 3.

Similarly, retail employment is located vis-a-vis people's residential choice. The probability that a shopping center will be located at zone 2, given a residential location at zone 1, is given by this formula

$$\frac{E_2^R/d_{12}^2}{E_1^R/d_{11}^2 + E_2^R/d_{12}^2 + E_3^R/d_{13}^2}$$

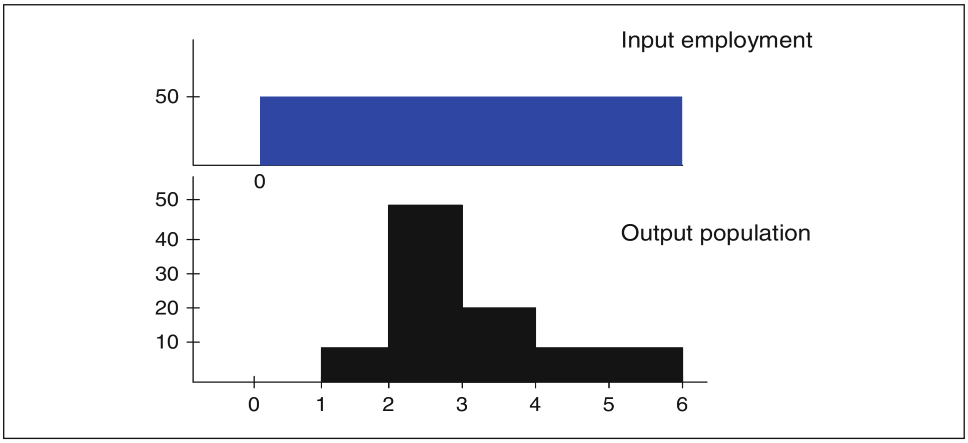
Problem 2: Further Discussions on Forecasting

It is commonly observed that population migration follows employment, albeit with a time lag. Here, we wish to estimate the distribution of population over the next six time periods following the introduction of employment (Chan 2005). The following shows the time lag for the dependent population to move into town:

- □ 0% of the population moves in during period 1 when employment is made available,
- □ 10% of the population moves in during period 2,
- □ 50% in period 3,
- □ 20% in 4,
- □ 10 in 5, and
- □ the remaining 10 in 6.

Such a time-lag relationship is also shown graphically below, where the number of jobs and the population size are expressed in ten's. For example, 50 jobs as shown in the top graph means actually 500 jobs, and a population of 10 actually means 100 in the lower graph (Figure 2.35):

Figure 2.35 INPUT–OUTPUT RELATIONSHIP



While 500 jobs are introduced in time period 1, subsequent jobs are available in varying quantities—300 in period 2, 900 in period 3 and so on. The same time-lag distribution is followed for subsequent employment introductions, as shown in the following Table 2.6. From the Table 2.6, it is clear that the employment can be gleaned from the bottom row. Each introduction of employment triggers in-migration of population, following the given time-lag distribution. The row sums amount to the total population in the study area for each time period, which are summarized in the right-most column in the Table 2.6. The following Table simply shows the employment and population series side-by-side, as extracted from the master Table 2.6 above:

Table 2.6 GROWTH IN ECONOMIC ACTIVITIES

Employment and population over time

period	emp*	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	pop*
1	5																					0
2	3	5																				5
3	9	25	3																			28
4	6	10	15	9																		34
5	5	5	6	45	6																	62
6	7	5	3	18	30	5																61
7	3		3	9	12	25	7															56
8	4			9	6	10	35	3														63
9	8				6	5	14	15	4													44
10	7					5	7	6	20	8												46
11	8						7	3	8	40	7											65
12	1							3	4	16	35	8										66
13	3								4	8	14	40	1									67
14	3									8	7	16	5	3								39
15	7										7	8	2	15	3							35
16	3											8	1	6	15	7						37
17	6												1	3	6	35	3					48
18	4													3	3	14	15	6				41
19	1														3	7	6	30	4			50
20	3															7	3	12	20	1		43
	emp	50	30	90	60	50	70	30	40	80	70	80	10	30	30	70	-	-	-	-	-	

*Figures are shown in units of 10's. For example, 50 jobs (instead of 5 jobs) are introduced to the study area in period 1.

period	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
emp	50	30	90	60	50	70	30	40	80	70	80	10	30	30	70	–	–	–	–	–
pop	0	5	28	34	62	61	56	63	44	46	65	66	67	39	35	37	48	41	50	43

Repeat the calculations for the same employment series, except that the time-lag distribution is now changed to

- □ 0% of the population moves in during period 1,
- □ 20% of the population moves in during period 2,
- □ 40% in period 3,
- □ 25% in 4,
- □ 15 in 5, and
- □ the remaining 10 in 6.

ENDNOTES

[1] In Section VIII of Chapter 4, we will discuss how to measure efficiency using Data Development Analysis, which is based on non-dominated solutions to more than one cost criterion.

[2] A full explanation of Hoyt's theory is given in Section II.B of Chapter 6.

[3] Figure 2.10 shows the tradeoff between the quantity of transportation and housing consumed in an indifference curve. On an indifference curve, a family sacrifices travel for better housing or vice versa for a given income. The tangency of the income/budget straight line and the indifference curve is the consumption level of the family.

[4] The series expansion for ln $(1 + x)$, where $-1 \leq x \leq 1$, is $x + x^2/2 + x^3/3 + \cdots$

[5] Much of the discussion in this section is taken from Vickerman (1980).

[6] These two cases will be analyzed in detail in later chapters when we construct models of market equilibrium.

[7] Recursive programming is explained in Appendix 3. Chan (2005) also illustrated application of recursive programming in his "Location-Routing Models" chapter. A software example is included on the attached CD/DVD under the RISE folder.

[8] Chan (2005) discussed alternate ways to effect this reallocation in his "Lowry-based Models" and "Bifurcation and Disaggregation" chapters. The readers may also wish to experiment with the software on the attached CD/DVD under the LOWRY and YI-CHAN folders.

[9] The answer to this module is attached at the end of this text book.

REFERENCES

Ahituv, N.; Berman, O. (1988) *Operations manual of distributed service networks—A practical quantitative approach*. New York: Plenum Press.

Alonso, W. (1964). *Location and land use: Toward a general theory of land rent.* Cambridge, Massachusetts: Harvard University Press.

Alonso, W. (1970). "Equilibrium of the household." In *Urban analysis: Readings in housing and urban development,* edited by A. N. Page and W. R. Siegfried. Glenville, Illinois.: Scott, Foresman and Co., 168–177.

Alonso, W. (1960). "A theory of the urban land market." *Paper and Proceedings of the Regional Science Association* 6:149–157.

Anderson, S. P.; Never, D. J. (1991). "Cournot competition yields spatial agglomeration." *International Economic Review* 32:793–808.

Bonsall, P. W.; Shires, J. D. (2006). "Employers' expectations for commuting & business-related travel in an ICT-Rich environment." *Transportation Research Record*, 1977:268–276.

Chapin, F. S.; Kaiser, E. J. (1979). *Urban land use planning,* 3rd ed. Urbana, Illinois.: University of Illinois Press.

Chan, Y. (2005). *Location, transport and land-use: Modelling spatial-temporal information.* Berlin and New York: Springer.

Cochrane, R. A. (1975). "A possible economic basis for the gravity model." *Journal of Transport Economics and Policy* 9:34–49.

Dahlman, C. J. (1988). "The problem of externality." In *The theory of market failure,* edited by T. Cowen. Fairfax, Virginia: George Mason University Press, 168–177.

de la Barra, T. (1989). *Integrated land use and transport modelling: Decision chains and hierarchies.* Cambridge, England: Cambridge University Press.

Dickey, J. W. (1983). *Metropolitan transportation planning,* 2nd ed. New York: McGraw-Hill.

Dorau, H. B.; Hinman, A. G. (1928). *Urban land economics.* New York: The Macmillan Co.

Friedrich, C. J., ed. (1929). *Alfred Weber's theory of the location of industries.* Chicago: Chicago University Press.

Hakimi, S. L. (1990). "Location with spatial interactions: Competitive locations and games." In *Discrete location theory,* edited by P.B. Mirchandani and R. L. Francis. New York: Wiley-Interscience, 439–478.

Hakimi, S. L. (1964). "Optimal location of switching centers and the absolute centers and medians of a graph." *Operations Research* 12:450–459.

Huff, D. L. (1962). Determination of intraurban retail trade areas. Real Estate Research Program. University of California at Los Angeles. Los Angeles.

Jha, K. Demographic models. Working Paper. Department of Civil Engineering. Pennsylvania State University. University Park, Pennsylvania.

Krueckeberg, D. A.; Silver, A. L. (1974). *Urban planning analysis: Methods and models.* New York: Wiley-Interscience.

Lakshmanan, T. R.; Hansen, W. G. (1965). "A retail market potential model." *Journal of the American Institute of Planners* 31, No. 2:134–143.

Leontief, W. W. et al. (1953) *Studies in the structure of the American economy*. New York: Oxford University Press.

Lund, J. R.; Mokhtarian, P.L. (1994). "Telecommuting and residential location: Theory and implications for commute travel in the monocentric metropolis." *Transportation Research Record* 1463, pp. 10–14.

Mai, C-C. (1986). "Random input transport rate and optimum location of the firm." *Transportation Planning Journal* (Taiwan, Republic of China) 18:311–331.

Marsh, M. T.; Schilling, D. A. (1994). "Equity measurement in facility location analysis: review and framework." *European Journal of Operations Research*, 74, No. 1:1–17.

Mokhtarian, P. L.; Meenakshisundarum, R. (1999). "Beyond tele-substitution disaggregate longitudinal structural equation modeling of communication impacts." *Transportation Research* 7:33–52.

Mokhtarian, P. L.; Varma, K. V. (1998). "The trade off between trips and distance traveled in analyzing the emissions impacts of center-based telecommuting." *Transportation Research* 3:419–428.

Nelson, J. (1971). "An interregional recursive program model of production, investment, and technological change." *Journal of Regional Science* 11: 33–47.

Newman, E. E. (1972) Economic concepts and models. Working Paper. Department of Civil Engineering. Pennsylvania State University. University Park, Pennsylvania.

Papageorgiou, Y. Y. (1990). *The isolated city state: An economic geography of urban spatial structure.* New York: Routledge (Chapman and Hall).

Van Lierop, W. (1986). *Spatial interaction modelling and residential choice analysis.* Hants, England: Gower.

Vickerman, R. W. (1980). *Spatial economic behaviour: The microeconomic foundations of urban and transport economics.* New York: St. Martin's Press.

Yeates, M.; Garner, B. (1980). *The North American city,* 3rd ed. San Francisco: Harper and Row.

3

Descriptive Tools for Analysis

"Most of the fundamental ideas of science are essentially simple and may, as a rule, be expressed in a language comprehensive to anyone."
Albert Einstein

A distinction is made in our discussion between descriptive versus prescriptive analysis techniques. A **descriptive model** is one that replicates the location and land use decisions made in a study area, while a **prescriptive model** starts out with a premise of existing practice and concentrates on the steps to arrive at a recommended course of action. Put in another way, a descriptive model summarizes the set of observed data and tries to explain these observations in a systematic manner. A prescriptive model, on the other hand, takes the view that the model has been constructed, and the model is used to choose a desirable course of action. In Chapters 3 and 4, we will review these analysis tools, paving the way for further analyses. The discussions here are geared toward problem solving; the development is therefore more intuitive than algorithmic or axiomatic in nature. We supplement these discussions with more methodological background materials attached as appendices of this book, where the readers will find self-contained reviews on optimization, stochastic process, statistics, and systems theory.

I. AN EXAMPLE

Three cities in Ohio—Cincinnati, Columbus, and Dayton—are planning a regional airport for their residents. By pooling resources, these cities will obtain a superior facility not possible without such cooperation. It is postulated that such an airport will have to be contained within the triangle defined by the three cities (Hurter and Martinich 1989). Within this triangle, it is not clear where the best location should be. The reader may recognize this as a Weber problem, as introduced in Chapter 2. If all three cities are equally important, one approach is to locate the airport in a central point convenient to all three cities. Such a location may be the

Y. Chan, *Location Theory and Decision Analysis*, 2nd ed., DOI 10.1007/978-3-642-15663-2_3,

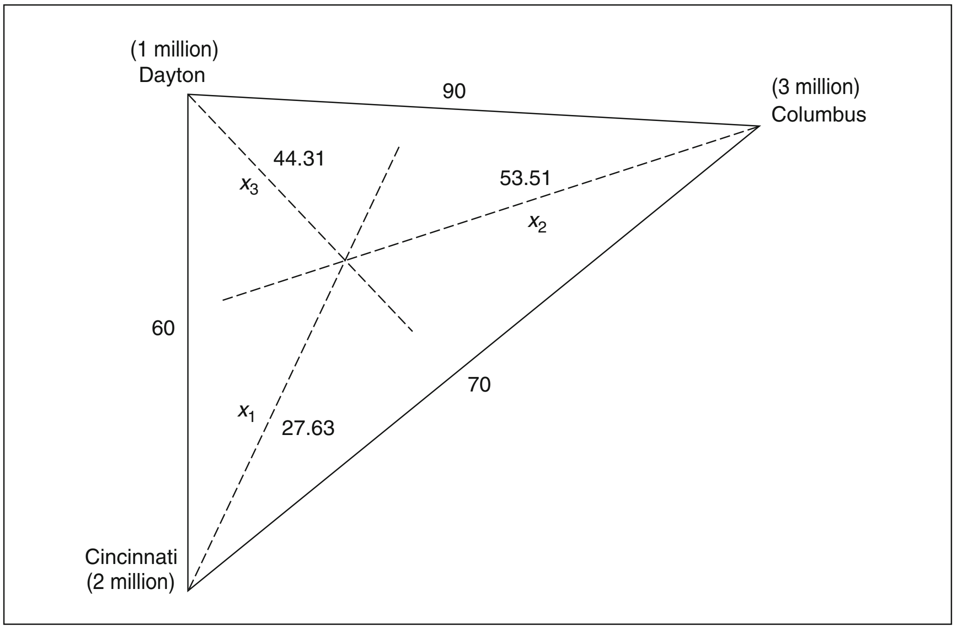
common point of the angle bisectors to the triangle defined by the three cities, as illustrated in Figure 3.1. The total travel time among the three cities is 27.63 + 53.51 + 44.31 = 125.46 minutes. This reflects the airport location most convenient for any citizen, irrespective of where he or she lives.

What happens to the location if each city is weighted differently, or if the siting decision is made by a central authority that has the aggregate interest of the entire region in mind? Assuming 125 minutes is the optimum, one would expect that the total travel distance will be larger than 125, since parochial interests—interests that stand in the way of the common good—are now taken into consideration depending on the weight placed on each city. Similar to the bisector case, each city pair would have combined airport travel times longer than the straight line between them, inasmuch as the airport may be located toward the third city, rather than along the corridor between the city pair concerned. Based on this reasoning and the above calculations, we can describe the candidate airport location more precisely as a set of inequalities:

$$\begin{aligned}&x_1 + x_2 \geq 70\\&x_1 + x_3 \geq 60\\&x_2 + x_3 \geq 90\\&x_1 + x_2 + x_3 \geq 125\\&x_1, x_2, x_3 \geq 0\end{aligned}$$

The above set of equations is by no means the only, nor is it necessarily the best, description of this location problem. An advantage of this descriptive model is its simplicity, which allows the construction of a prescriptive model by superposing

Figure 3.1 A LOCATION DETERMINED BY ANGLE BISECTORS



SOURCE: Claunch, Goehring, and Chan (1992). Reprinted with permission.

an objective function such as minimizing the person-minutes-of-travel: Min $2x_1 + 3x_2 + x_3$, where the set of weights are the population sizes on each city. This example, simple as it may be, may bring out the difference between what we mean by a descriptive versus a prescriptive model.

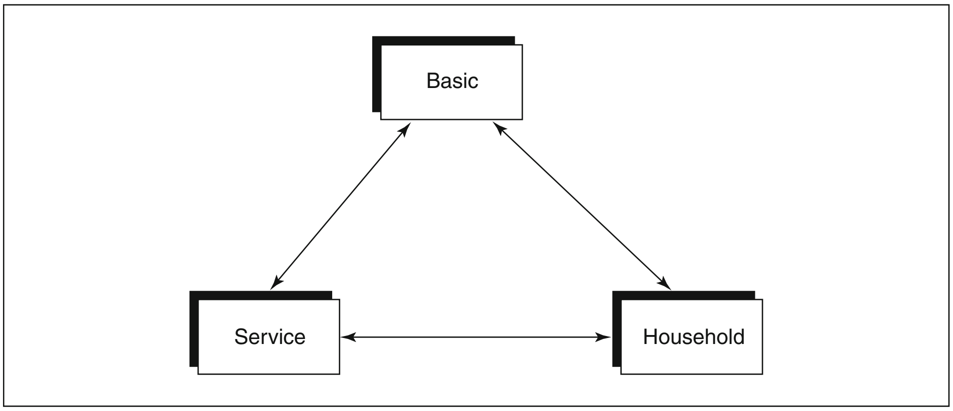
II. DESCRIPTIVE TECHNIQUES: ANOTHER EXAMPLE

In the example cited in Chapter 1, the relationship between basic and secondary economic activities in New York City and New Haven is identified, and a model is built to reflect the observed phenomenon. The relationship can often be displayed graphically in a flow chart as a first step of the analysis. To operationalize the flow chart, parameters such as the average size of a family in the study area need to be calibrated. In Chapter 1, we have already sketched out a flow chart entitled "Economic Interaction between New York and New Haven Over Time" and assumed some calibration parameters. The generic term, descriptive techniques, is used to include logical flow charts and calibration. Such tools are the ways and means to construct a model replicating the study area.

Generally speaking, there are six steps in building a descriptive model. Again using the New York-New Haven development example,

Step 1: identifies the system components. In this example, there are three economic sectors: basic, service, and household. They are related in a pairwise manner in Figure 3.2. Basic employment refers to the new jobs introduced to either New York or New Haven. The household sector initially encompasses all the dependents of the workers brought into the area. Correspondingly, service employment consists of additional jobs required to support the households that are now located in the area.

Figure 3.2 BLOCK DIAGRAM OF THE NEW YORK-NEW HAVEN DEVELOPMENT EXAMPLE



Step 2: follows with a definition of what goes into these economic sectors. One may define, for example, all support services including such services as medical, fire, education, shopping, food, and entertainment, under a single service sector rather than seperating them into the public service sector (e.g., medical, fire and education) and the private service sector (e.g., shopping, food, entertainment).

Step 3: defines specific variables corresponding to the sectors or subsystems. For example, one needs to define the basic employment as E^B, service employment E^R, and population N. Attention is paid to geographic or spatial attributes too, such as the population and employment in New York versus New Haven. Finally, we distinguish the activities in the base year versus the forecast year—in which case a temporal attribute is associated with the variables. In this example, five time periods were modeled.

Step 4: delineates the mechanisms of change or casual structure of the system. Using the New York-New Haven example again, basic employment is isolated as the seed for other dependent developments, such as service employment and population. Furthermore, the population and employment in New York and New Haven interact with each other, as shown in the commuting pattern between the two cities, in which the population of one city may find employment in another.

Step 5: One decides between a descriptive versus prescriptive application, that is to say, whether the model is to be used primarily to answer what-if type questions or in plan specification. A descriptive formulation strives to capture the development pattern of the study area as a primary focus. For a prescriptive formulation, on the other hand, specific goals and objectives of the community need to be explicitly modeled on top. In the New York-New Haven example, a descriptive, rather than prescriptive, model is constructed.

Step 6: One assembles all the aforementioned elements into a coherent model. This means the variables defined in *Step 3* are related to each other in a set of equations or other mathematical framework relating the logical structure identified in *Step 4* and in light of the application intent of *Step 5*. In subsequent discussions, we will see how this is accomplished for the New York-New Haven example.

There will be many occasions in an analysis professional's career when a model, whether descriptive or prescriptive, needs to be constructed. The above six steps will become a handy checklist for model building. This chapter will focus on descriptive techniques. These types of analysis tools will be discussed: simulation, queuing, econometrics, and calibration. They will be introduced in an order that parallels our discussions on model building. The sequence also starts with the less complex and progresses toward the more sophisticated.

III. SIMULATION

Perhaps no other tool can illustrate a descriptive model better than simulation, since a simulation model simply replicates the existing phenomenon in the study area. **Simulation** is a familiar analysis tool since it is easy to understand and apply. Notice this does not make the claim that people invariably apply it correctly; in fact, the contrary is true. There are more misuses than valid uses of simulation. It makes it so much more important, therefore, to put this analysis tool inperspective.

In the first stage of building a simulation model, components of the system and their interrelationships need to be identified. These interrelationships may be preliminary postulations that are subject to verification and validation in later stages. The basic components of a system are best displayed in a block diagram. We have already discussed the example illustrated in Figure 3.2. In this figure, the interdependency among the basic, service, and household sectors is shown by the use of arrows. Thus in a visual manner, one can see these economic sectors are tied together. The next step in this type of descriptive modeling involves a logic flow chart. The flow chart details the aggregate relationship identified in the block diagram, in which one examines the precise casual chain of events. Figure 3.3 illustrates such a chart for the New York-New Haven example. Basic employment generates dependent population. The population requires services, thus bringing in service employment. The service employees in turn have their dependent population brought into the area. Figure 3.3 clearly identifies basic employment as the seed of the subsequent activities. Furthermore, it highlights the cyclical generation of population and service employment.

Simulation can be deterministic or stochastic. **Deterministic simulation** can be best described as the modeling of the average condition of the system, ignoring the transient and time-varying behavior. **Stochastic simulation,** on the other hand, specifically gears toward the random fluctuation of the system. The New York-New Haven example is a good illustration of deterministic simulation. If we ignore the spatial interaction between the two cities, a simple model can be constructed. Suppose the variables are: basic employment

Figure 3.3 LOGIC FLOW CHART OF THE NEW YORK-NEW HAVEN EXAMPLE

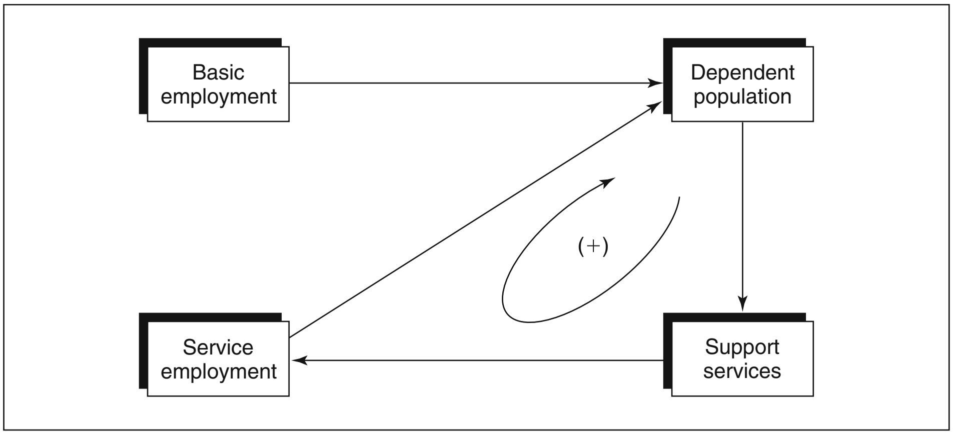


Table 3.1 ILLUSTRATION OF A DETERMINISTIC SIMULATION ON REGIONAL ECONOMIC DEVELOPMENT

Time n	Basic emp E^B	Basic emp pop	Support service emp E^R	Support service pop	Total emp E	Total pop N
1	E^B	fE^B	afE^B	af^2E^B	$(1+af)E^B$	$f(1+af)E^B$
2			$a^2f^2E^B$	$a^2f^3E^B$	$(1+af+a^2f^2)E^B$	$f(1+af+a^2f^2)E^B$
3			$a^3f^3E^B$	$a^3f^4E^B$	$(1+af+a^2f^2+a^3f^3)E^B$	$f(1+af+a^2f^2+a^3f^3)E^B$
…			…	…	…	…
∞			0	0	$(1+af+a^2f^2+a^3f^3+\cdot\cdot)E^B$	$f(1+af+a^2f^2+a^3f^3+\cdot\cdot)E^B$

E^B, population N, average household size f, and service employment multiplier a (defined as the number of service jobs generated from one resident.) The economic development of either New York or New Haven (without commuting between the two cities) can be modeled by Table 3.1. which is basically a quantification of the numerical calculations documented in Table 1.1 in Chapter 1 in which the increase in basic employment generates subsequent development through multiplier effect. The reader may wish to view the two tables side by side for comparison purposes.

It can be seen that such a procedure points toward an employment increment of a^nf^nEB and a population increment of $a^nf^{n+1}E^B$ in the nth iteration. In other words, each iteration through the loop between dependent population, support services, and service employment in Figure 3.3 generates another increment of activities. As iterations progress, which can be thought of as time progresses in this example, the amount of activities generated can become smaller and smaller, remain the same, or increase, depending on the household size f and the population serving ratio a. Thus the simulation will either yield dampened growth or unlimited growth depending on the total employment series $(1 + af + a^2f^2 + a^3f^3 + \cdots)E^B$ and the total population series $f(1 + af + a^2f^2 + a^3f^3 + \cdots)E^B$.

Recognizing the similarities to a geometric series

$$1 + x + x^2 + x^3 + \cdots = \frac{1}{1-x} \qquad (x < 1), \tag{3.1}$$

the series above have analytical solutions if $af < 1$, where the total employment series sums up to $E^B/(1-af)$ and the total population series sums up to $fE^B/(1-af)$. Thus in the New York-New Haven example in Chapter 1, where $f = 2.5$ and $a = 0.2$, the total employment is $1000/[1 - (0.2)(2.5)] = 2000$, and the total population is $(2.5)(2000) = 5000$. The series will have no immediate closed form solution for $af = 1$ and $af > 1$. Thus simulation is a more versatile tool than analytical solutions in general, in that simulation can provide a solution where analytical methods fail.

Another example of deterministic simulation is the **limits to growth model** (Meadows et al. 1972). Following the procedures of this model, the feedback loop flow diagram of Figure 3.3 can be illustrated by using a positive sign, meaning that there is a reinforcement effect among the variables, as is typical of the multiplier effect in a regional economy. In a world forecasting model, Meadows et al. constructed a flow diagram consisting of both positive and negative feedback loops, depicting the interactions among the various sectors of the world economy. For example, population is positively related to the birthrate, and negatively related to the death rate, meaning that an increase in the birthrate will further increase the population, whereas an increase in the death rate will accelerate a decline in the population:

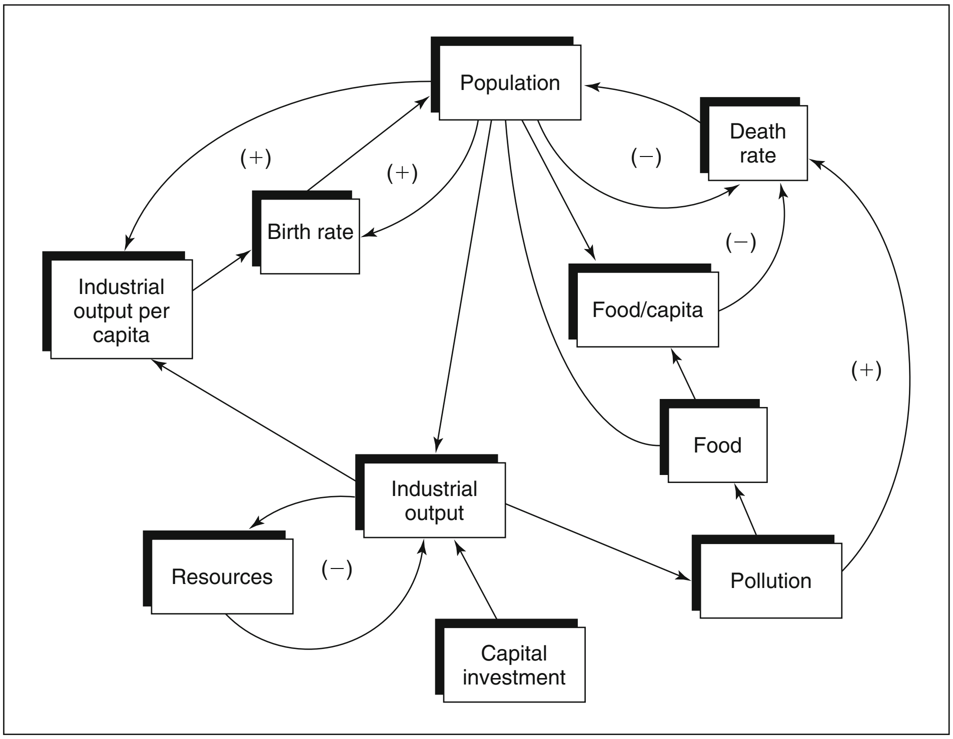
$$POPULATION = F_1(BIRTH\ RATE,\ DEATH\ RATE)$$

Birth- and death rates are again dependent upon the industrial economy well-being, agricultural food production, and the environmental condition:

$$BIRTH\text{-}DEATH\ RATE = G(INDUSTRIAL\ OUTPUT\ PER\ CAPITA,\ FOOD\ PER\ CAPITA,\ POLLUTION)$$

In a similar manner, the feedback loops in Figure 3.4 can be represented in the remaining set of equations:

Figure 3.4 LIMITS TO GROWTH MODEL



INDUSTRIAL OUTPUT = F_2(*CAPITAL INVESTMENT, POPULATION, RESOURCES*)

POLLUTION = *POLLUTION OF PREVIOUS YEAR* +
(*POLLUTION-GENERATION RATE*) (*INDUSTRIAL OUTPUT*)

RESOURCES = *FIXED AMOUNT* − (YEARLY RATE OF USE) (*INDUSTRIAL OUTPUT*)

FOOD = F_3(*POLLUTION, POPULATION*)

Clearly, this logic-flow diagram is much more complex than the economic-base model for New York-New Haven. The relationship needs to be expressed in the language of the computer to make the model feasible for use. In the following computer run, the inputs to the model consist of:

birthrate = 0.035 (or 35 in 1000) per year

death rate = 0.015/year

food production rate = 1 percent growth per year

resource use rate = 0.4 percent/year

industrial output rate = 4 percent/year

pollution generation rate = 0.3 percent/year.

Starting out with these world statistics in 1970:

population (P) = 3.6 billion

food-production rate (F) = 1 unit/person (the average of 2000 calories per person per day)

world resource (R) = 1 unit

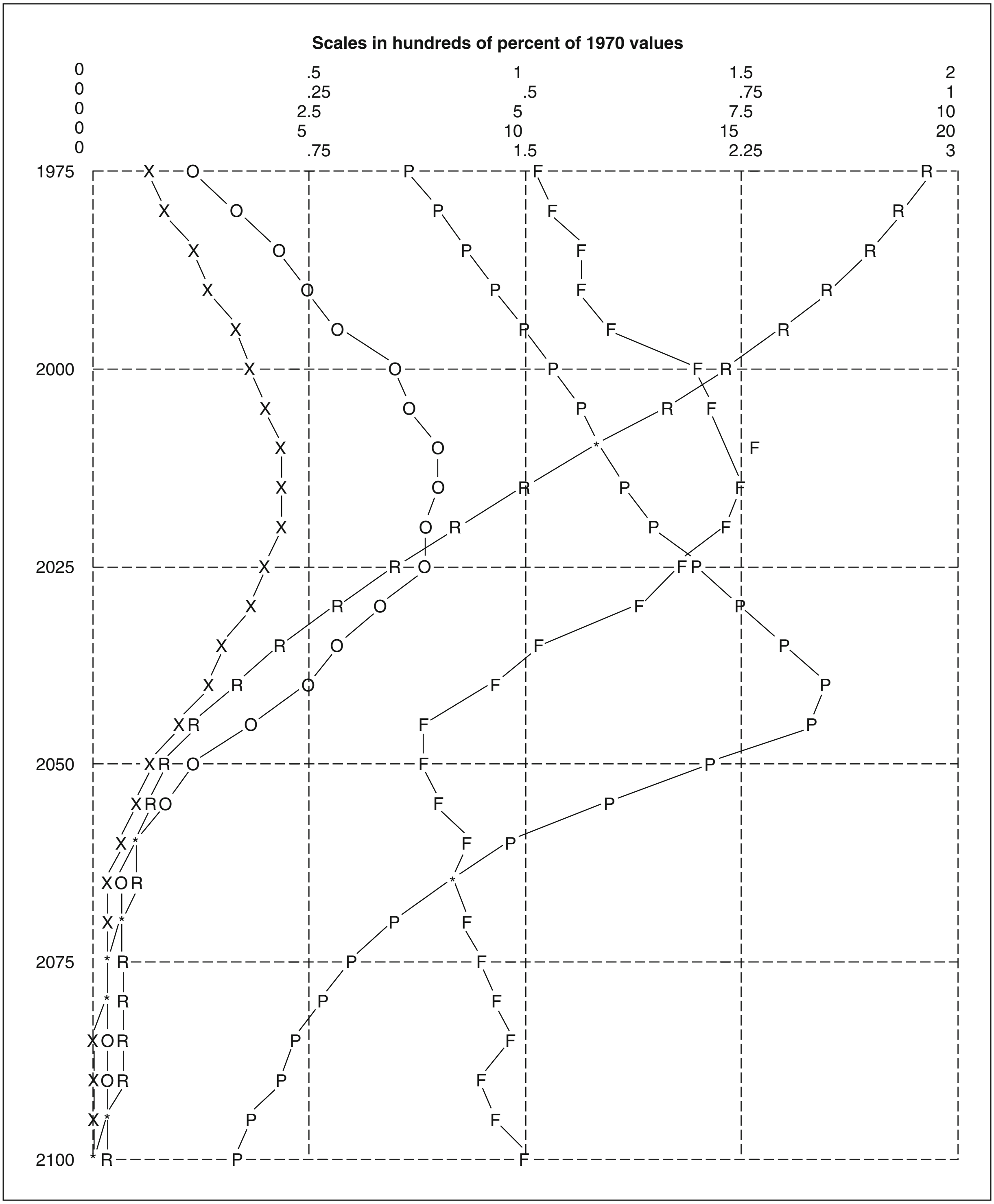
industrial output (O) = 1 unit (equivalent to 2 trillion dollars)

pollution generation rate (X) = 1 unit.

The model in turn forecasts these statistics in the world through the year 2100. The highly publicized doomsday result is sketched in Figure 3.5. After some transient phenomena, the steady state condition is reached where resources are depleted and the world population dwindles to a few.

Other global simulation models include World Integrated Model, Latin American World Model, United Nations Input-Output World Model, and Global 2000, just to name some of the major ones (Congressional Office of Technology Assessment 1982). Worthy of note is that spatial elements are totally absent in all these models, including the limits to growth model. In other words, the entire world is treated as an entity and one does not distinguish between the continents, countries, states/provinces, and regions. We will see how such deficiencies can be redressed in subsequent discussions.

Figure 3.5 WORLD FORECAST THROUGH 2100



IV. STOCHASTIC SIMULATION

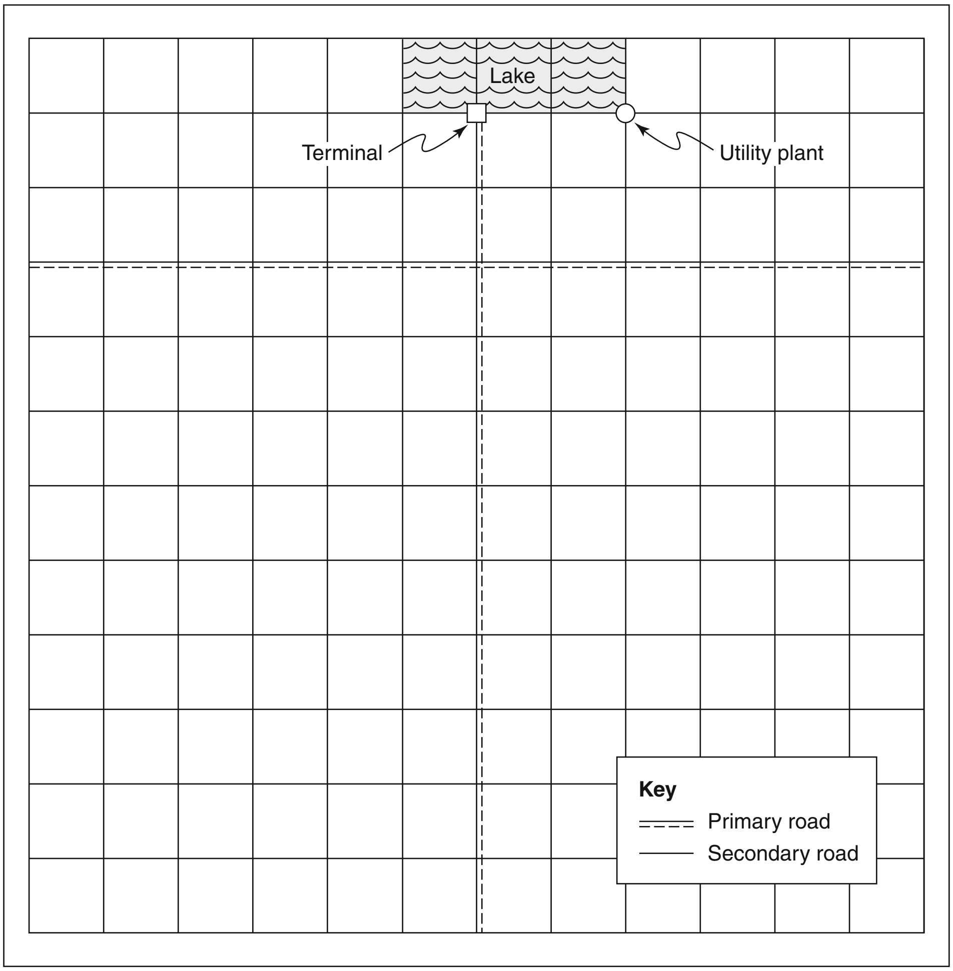
While deterministic simulation serves as a good introduction, much of simulation modeling involves uncertainty, which is an integral part of the model. Stochastic simulation has its Monte Carlo variety and discrete event variety. Here, we define **Monte Carlo simulation** as the model which addresses uncertainty through random number generators, which effectively generates probability distributions through much the same idea as a roulette wheel. Thus the probability of certain events taking place is determined by spinning such a roulette wheel. A clock may be used to keep track of time increments, as was done in the limits to growth model. At each time increment, an event may or may not happen depending again on the random number generator that serves as the roulette wheel in the computer. Discrete event simulation, on the other hand, goes one step further in sophistication. It goes from the current event to the next event in sequence, with the clock updated as it processes the next event. Most people associate this branch of simulation with computer languages, including GPSS, SIMSCRIPT, GASP, SLAM (Pritsker 1986) and SIMAN. Recent advances in computer science call for object-oriented simulation languages that allow for model execution efficiency. Discrete event simulation has not been as widely used as regular Monte Carlo simulation in facility location and land use, simply because of the more aggregate nature of location decisions. Recent requirements of a service economy, however, have changed this practice dramatically, as we will see in Section V.

The best way to illustrate Monte Carlo simulation is still through examples. Many land use games are used to introduce students of planning to the many political and institutional factors in development. One of these is the **Community Land Use Game (CLUG)** developed by Feldt (1972). The decision-making process on urban development is often characterized by conflicting interests seeking social, political, and financial gain. Short of actual experience (which sometimes turns out to be costly), the use of games is one of the best ways to highlight the issues. CLUG is usually played by three or more teams, each of which consists of two or more members. A community is represented on a square with a grid of secondary road network and spines of primary roads. A utility plant, denoted by a circle, is set up, but without any distribution and collection facilities (see Figure 3.6). The game board represents the site of a community yet to be developed by the players. The local economy is connected with the outside world through a transportation terminal (marked as a square at the waterfront). The game simulates the development of a brand new community, as catalyzed by the initial location of basic industries (or export service industries). In other words, external investment starts the development of the local economy.

Parallel to the concept of the economic-base theory, there are three economic sectors represented in the game:

1. **Basic industries,** consisting of full industries (FI) and partial industries (PI);
2. **Residential sector,** where the housing density ranges from sparse to dense, as represented by single residence (R1), double residence (R2), triple residence (R3), and quadruple residence (R4);
3. **Service sector,** which is exemplified by the central store (CS), local store (LS), and office unit (O). Each of these economic units is characterized by its construction cost, income, number of employees,

Figure 3.6 THE CLUG PLAYING BOARD



SOURCE: Feldt (1972). Reprinted with permission.

payroll, service costs and so forth. Corresponding to these sectors are also the land-use parcels (represented as a cell or square on the board) on which development can take place.

Aside from these players, someone serves as a city manager, who represents the community to the outside world, particularly in its financial management. He is the exogenous party who constructed the transportation system connecting the community with the outside world, thus laying the groundwork for future developments in the local community. The community develops the industries that occupy the land *ab initio* and that provide a tax base for the construction of the utilities. The industries, in turn, employ the labor forces from the residential population. In accordance with the economic-base theory, secondary activities such as the retail services are attracted into the city. The monetary flow between these sectors serves as a link between the various activities, and the community pays for their public utilities and other public services through taxes.

The development sequence is simulated by the following steps in the game:

1. **Purchase land.** Each team is an entrepreneur from the outside world who is contemplating land purchase investment in the study area. A bidding process is conducted, and the highest bidder on a parcel of land will be awarded its ownership.
2. **Provide utilities.** Since only the power plant exists in the beginning, power lines have to be constructed before further development is possible. Power lines have to be provided for at least one side of a parcel before that land can be developed. A majority consensus de-termines the precise location of a line. The construction of utility systems is to be paid for from the general tax coffers of the community.
3. **Renovate.** If this is the second or higher round of the game, the study area would have been built up already, consisting of a number of existing buildings. During the useful life of an existing building, there is a chance that the building will be lost, and the chance increases with age. Table 3.2 determines the probability of loss via the use of a pair of dice or a random number generator. For example, when the building is 10 years old, it is condemned if the equivalent numbers on a pair of dice are 5, 6, 7, or 8, which is equivalent of a loss probability of 0.556. In the event that a building is condemned, it can be demolished during the construction step of any round, in other words, at the next step.
4. **Construct building.** New construction is now considered on all lands owned by an entrepreneur, as long as utilities are available for the site. Two chances are given to each team to either construct or pass. Any construction decision should weigh income potential against cost.

Table 3.2 PROBABILITY OF BUILDING LOSS

Age of building	Probability of loss	Losing numbers
0	0.056	3
1	0.111	5
2	0.167	7
3	0.195	2, 7
4	0.250	2, 7, 11
5	0.306	2, 7, 9
6	0.362	3, 7, 8
7	0.417	5, 7, 8
8	0.445	6, 7, 8
9	0.500	3, 6, 7, 8
10	0.555	5, 6, 7, 8

5. **Designate employment.** A contractual agreement is to be made between the industry/service sectors and the household sector about the commitment of labor forces to employment opportunities. Thus both the full and partial industries, stores, and the offices bid on the existing labor pool, culminating in contracts being signed.
6. **Set prices in local store (LS), central store (CS) and office (O).** An arrangement is to be worked out between the LS and CS with the residential sector about the price of goods. Similarly, contracts are signed between the office/industries and stores in regard to the purchase of goods and services from one another. While the unit price can be negotiated with the stores and offices in the community, a fixed price is charged for goods and services purchased from the outside world through the city manager.
7. **Receive income.** In order to start the process, some incentives have to be provided to the industries to start production. The city manager gives income to the industries for putting people to work, a process simulating receiving gross earnings from the manufactured goods. The income is set above the wage rate in order to show a profit margin above labor cost. This is the only money paid by the city manager to the players; all other income is generated through payments among teams for payrolls, payments to stores, and so forth.
8. **Pay employees.** Each team owning a land use which employs people from residential units owned by another team pays that team a labor wage.
9. **Pay LS, CS and O.** Upon completing the exchange of payments for meeting payrolls to employees, each team owning residential units must make payments to the local and central stores with which they are trading at the agreed-upon price. Notice that each residential unit must make payment to two kinds of stores, both local and central, corresponding to the two types of market baskets purchased. Similarly, offices are paid for the services rendered.
10. **Pay transportation.** For each industry, the players compute the cost of shipping to the terminal by counting the number of units of distance traveled, distinguishing the different unit costs between a major highway and a secondary road. If the industry is shopping at an office on the board, the weighted distance to this unit should be computed similarly. Then players take each residential unit in turn and compute their transportation costs to work or shop also. Finally, they finish off with the LS or CS, who also use the roads to deliver their goods. These figures are then summed to yield the total transportation cost for each team.[1]
11. **Pay taxes.** Tax is levied against the real estates as a percentage of the respective assessed values. Charges are also made against the community for construction of new utility lines, for maintenance of old lines, and for social services for each residential unit. The comparison of this cost to total taxes raised can be shown to yield the community surplus or deficit for the current round. When an individual team cannot meet its tax obligations, the city manager will begin foreclosing until sufficient value has been received to meet the required tax debt.

These 11 steps illustrate the parallel between CLUG and the real world. Particularly worthy of note is the renovation portion of the simulation, where the probabilistic statement of the simulation comes in. This example shows graphically the use of Monte Carlo simulation in land use games and the similarity between certain aspects of gaming and simulation.

V. DISCRETE EVENT SIMULATION

To properly understand discrete event simulation, some background on queuing and time-dependent random process (stochastic process) is necessary.[2] Suppose a fleet of vehicles is responding to service requests in a network. A model developed by Larson, as cited by Ahituv and Berman (1988), aids in assessing the system performance under normal operating conditions (or under steady state of the system). **Larson's hypercube model** assumes that demands can be represented by different independent point sources. The point sources are represented by a **centroid,** a regular node in the network where all demands around the vicinity are supposed to originate. Calls for service arrive at the centroid according to a **time-homogeneous Poisson process,** a random pattern that averages at a given arrival rate λ'':

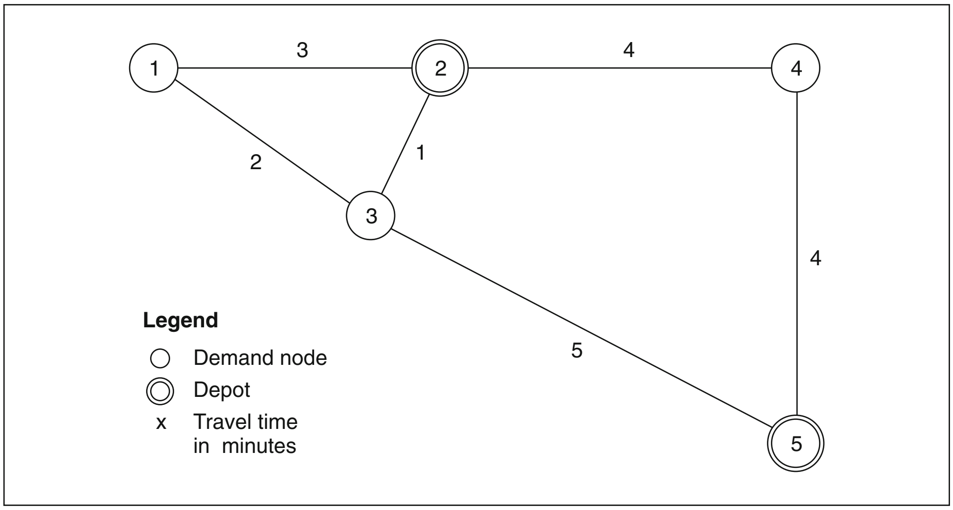
$$P(k) = \frac{\lambda''^{k} e^{-\lambda''}}{k!} \qquad k = 0, 1, 2, \ldots \tag{3.2}$$

where k is the random number of actual demand-requests per unit-time. The service time τ, setting aside the enroute travel time, for each vehicle unit is assumed to be negative exponentially distributed—again a random process with a given mean $1/\mu'$: $f_\tau(\tau) = \mu' e^{-\mu'\tau}$, where τ denotes the random variable for service time.

A. Stochastic Process

Each vehicle server, say a fire truck that helps to put out a fire, may be in two possible states, busy or free. When a call arrives, a single vehicle unit is chosen from those that are free and is immediately assigned to provide service. In the event that all servers are busy, the call is either lost (in other words, passed to a nearby jurisdiction for service) or queued until a unit becomes available. We call the former **zero capacity queue** (or the **loss system model**) and the latter **infinite capacity queue.** The hypercube model provides a steady-state analysis as an approximation to time non-homogeneity. With the model, many performance measures of system effectiveness can be derived. Among the important measures are the expected service unit response time, service unit dispatch frequencies, service unit workload, and the workload of a particular unit relative to the other units. To demonstrate the model, refer to the sample network of Figure 3.7. We assume that service stations (depots) are located at nodes 2 and 5. At each station, there is only one vehicle. Whenever there is a call, the dispatcher will assign the closest available vehicle to serve the calling node. The dispatching center can assign only stationary vehicles while they are at their depots. The center cannot contact a moving vehicle. When all units are busy, a special service unit from another jurisdiction will be dispatched (assuming a zero capacity queue). In

Figure 3.7 SAMPLE NETWORK FOR HYPERCUBE MODEL



SOURCE: Ahituv and Berman (1988). Reprinted with permission.

addition, we know that calls for service are issued, on the average, every 1/3 minute ($\lambda'' = 3$). The average service takes 1 minute ($\mu'' = 1$). The length of the links is measured in time units. The server's speed of travel is constant, and the demand for service, *k*, is equally divided among the centroids (i.e., the fraction of demand among the centroids is $f_1 = f_2 = f_3 = f_4 = f_5 = 1/5$.)

Knowing that there is only one server vehicle at each depot, we may say that at any time a depot either possesses an idle vehicle or it does not. We will denote it by 0 or 1, respectively. A state of the system is defined to be a vector of two components. The first component indicates the status (free or busy) of the server at node 2, and the second component indicates the status of the server at node 5. Since there are only two depots in the example, the entire network can be in any of the following four states:

(0, 0) the two vehicles are available at both nodes 2 and 5;

(0, 1) only the vehicle from node 2 is available;

(1, 0) only the vehicle from node 5 is available;

(1, 1) no vehicle is available from either node 2 or node 5.

Now based on the shortest distance, we can devise a dispatching table for the network, recalling that the policy is always to dispatch the closest available unit and to do nothing if both vehicles are busy (ties can be broken arbitrarily). Table 3.3 describes the dispatching rules for each of the four states:—✓designates "dispatch," —designates "do not dispatch." It is easily seen from the table that the closest vehicle is dispatched whenever both vehicles are available [state (0, 0)]. In other states, there is only one dispatching possibility; thus, there is no dilemma as to which unit to dispatch.

Now that we have established the dispatching rules, we would like to investigate the process by which the network changes from one state to another.

Table 3.3 DISPATCHING RULES FOR THE HYPERCUBE NETWORK

State	**Server vehicle location**			**eman no e**		
		1	**2**	**3**	**4**	**5**
(0, 0)	2	✓	✓	✓	—	—
	5	—	—	—	✓	✓
(0, 1)	2	✓	✓	✓	✓	✓
(1, 0)	5	✓	✓	✓	✓	✓
(1, 1)		not relevant				

SOURCE: Ahituv and Berman (1988). Reprinted with permission.

For instance, when the network is in state (1, 0) at a certain time interval, it can either change to state (1, 1) if the vehicle at node 5 is assigned to a call, or it can enter into state (0, 0) if the vehicle stationed at node 2 has been released from a previous service call (and is back at node 2). The network cannot change directly from (1, 0) to (0, 1) owing to the Poisson arrival and negative exponential service. In other words, a transition from (1, 0) to (0, 1) would imply that two events can occur in a very short time interval *dt*—the vehicle at node 5 is assigned while the server at vehicle 2 is being freed at the same time. Figure 3.8 depicts the transitions from one state to another. By observing Figure 3.8, we can use the information about the service rate y_1 and the call rate λ'' to derive the rate of the various transitions. For instance, when the network is in state (0, 1), it will change to state (1, 1) at a mean rate of 3 per minute, because on the average there is a call every 1/3 minute. On the other hand, the transition from (0, 1) to (0, 0) is at a mean rate of 1 per minute, since the average service time is 1 minute. A similar computation can be preformed for state (0, 0). If a call arrives from node 1, 2, or 3, the vehicle from node 2 will be dispatched. Since the rate of calls is 3 per minute and nodes 1, 2, and 3 each assume one-fifth of the overall demand, the transition rate from state (0, 0) to state (1, 0) is

$$\left(\frac{1}{5} + \frac{1}{5} + \frac{1}{5}\right)\ (3) = 1.8$$

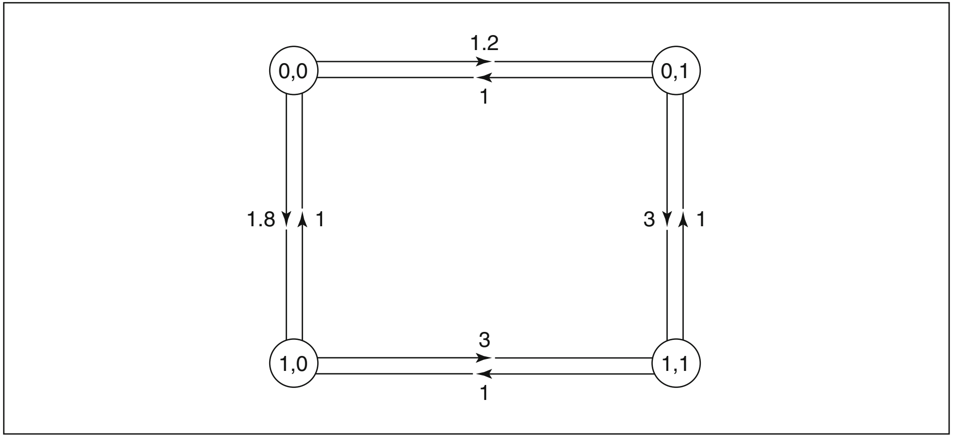
Similarly, the transition rate from (0, 0) to (0, 1) is

$$\left(\frac{1}{5} + \frac{1}{5}\right)(3) = 1.2$$

These transition rates are again summarized in Figure 3.8.

Now we assume the network is in balance (steady state); namely, it makes transitions from one state to another with a regularity that reflects an equilibrium between demand for and supply of services. This implies that there are

Figure 3.8 TRANSITION BETWEEN STATES IN A HYPERCUBE MODEL



SOURCE: Ahituv and Berman (1988). Reprinted with permission.

steady-state probabilities for being in the various states. We will denote them by $P_{(0,0)}$, $P_{(0,1)}$, $P_{(1,0)}$, and $P_{(1,1)}$. Equilibrium implies that the expected rate of leaving a state is equal to the expected rate of entering into the same state. For example, the expected rate by which the network departs from state (0, 1) is $3\ P_{(0,1)} + 1\ P_{(0,1)}$. The first term refers to a transition to (1, 1) while the second term refers to a transition to (0, 0). Similarly, the expected rate at which the network arrives at state (0, 1) is a weighted sum of all the transition rates from states that can transition into (0, 1); specifically, $1.2\ P_{(0,0)} + 1\ P_{(1,1)}$. Since in the steady state there should be a balance between the expected rates of entering and leaving a certain state, we may write a balance equation for (0, 1):

$$3\ P_{(0,1)} + 1\ P_{(0,1)} = 1.2\ P_{(0,0)} + 1\ P_{(1,1)}$$

Similarly, we can obtain balance equations for all the other states. For (0, 0), we will obtain

$$1.8\ \mathrm{P}_{(0,0)} + 1.2\ \mathrm{P}_{(0,0)} = 1\ \mathrm{P}_{(0,1)} + 1\ \mathrm{P}_{(1,0)}$$

For (1, 0), we will obtain

$$1\ P_{(1,0)} + 3\ P_{(1,0)} = 1\ P_{(1,1)} + 1.8\ P_{(0,0)}$$

For (1, 1), we obtain

$$1\ P_{(1,1)} + 1\ P_{(1,1)} = 3\ P_{(1,0)} + 3\ P_{(0,1)}$$

In addition to these balance equations, we know that the state probabilities should add to 1:

$$P_{(1,1)} + P_{(1,0)} + P_{(0,1)} + P_{(0,0)} = 1$$

Now we have five equations to find a solution for four unknown probability values. Any three of the balance equations and the last one will do:

$$\begin{array}{rrrrl}
-1.2\,P_{(0,0)} & +4\,P_{(0,1)} & & -1\,P_{(1,1)} & = 0 \\
3\,P_{(0,0)} & -1\,P_{(0,1)} & -1\,P_{(1,0)} & & = 0 \\
-1.8\,P_{(0,0)} & & +4\,P_{(1,0)} & -1\,P_{(1,1)} & = 0 \\
P_{(0,0)} & +P_{(0,1)} & +P_{(1,0)} & +P_{(1,1)} & = 1
\end{array}$$

The solution to these equations provides the steady-state probabilities for the network: $P_{(0,0)} = 0.1176$, $P_{(0,1)} = 0.1676$, $P_{(1,0)} = 0.1853$, and $P_{(1,1)} = 0.5295$. The results indicate that more than 50 percent of the time the two vehicles will be *busy* $[P_{(1,1)}]$. Around 11 percent of the time, the two servers will be idle $[P_{(0,0)}]$. The expected response time for this example is (where R' is the required time in dispatching a special reserve unit from a neighboring jurisdiction.)

$$P_{(0,0)}\left[\sum_{i=1,2,3} f_i d^{2,i} + \sum_{i=4,5} f_i d^{5,i}\right] + P_{(0,1)}\sum_{i=1}^{5} f_i d^{2,i} + P_{(1,0)}\sum_{i=1}^{5} f_i d^{5,i} + P_{(1,1)}R' \tag{3.3}$$

With $R'=10$ and all time separations d^{kl} obtainable from the shortest paths on the Figure 3.7 network, the above expression can be verified to be (0.1176)[(0.2)(3 + 0 + 1 + 4 + 0)] + (0.1676)[(0.2)(3 + 0 + 1 + 4 + 6)] + (0.1853)[(0.2)(7 + 6 + 5+ 4 + 0)] + (0.5295)(10) = 6.7677. The workload of the vehicle at node 2 is $P_{(1,1)} + P_{(1,0)} = 0.7148$ or this vehicle is busy about 71 percent of the time. The workload of the vehicle at node 5 is $P_{(1,1)} + P_{(0,1)} = 0.6971$. The fraction of dispatches that send the vehicle from node 2 to node 1 is $f_1[P_{(0,0)} + P_{(0,1)}] = (0.2)(0.1176 + 0.1676) = 0.0570$.

B. Simulation

The memory and execution time required to solve the hypercube model equations roughly doubles with each additional server. In other words, the procedure requires solving $2^{Q'}$ equations, where Q' is the number of servers. This can amount to huge computational requirements. Among ways to overcome this is discrete event simulation, which has become an efficient tool. This is made possible by the availability of very user-friendly simulation languages. Instead of solving simultaneous equations, the simulation processes each demand-request through the network. The term **discrete event** refers to examining the next event, whether it be another demand generated, a demand ready to be served, and so on. The program turns the clock on and eventually tallies up the performance statistics as computed analytically above.

It is not until recently that discrete event simulation has played a significant role in facility location. The City of Baltimore, for example, conducted a study to locate Emergency Medical Services (EMS). A validated discrete event, stochastic simulation model, Ambulance System Site Inspection Simulation Technique (ASSIST), was used to measure the performance of the EMS, pointing toward a city wide response time of 5 minutes; 95 percent of the demand was responded to within 10 minutes. Though these statistics indicated a well-run system, the EMS administrators requested a study because of perceived inequities in the spatial distribution of service, outlying areas tended to have higher response times than areas in the center of the city. ASSIST was used to validate the location of EMS by an optimization model (Heller et al. 1989). Parameter estimation of the optimization model was based on the same data used in the simulation and, where necessary, deterministic estimation of random variables. Emergency travel times were approximated in the simulation using two databases: (1) two travel time matrices, representing peak and off-peak (8-hour time intervals each) traffic conditions for travel between the activity centroids of any two nodes in the 207-node transportation network, and (2) about 4200 medic-unit run tickets that documented the spatial and geographic history of the response. The travel time parameters in the location model, τ_{ij}, were defined by the average of the estimated peak and off-peak emergency travel times.

EMS has been simulated as a non-homogeneous Poisson process, with arrival rates defined for 24 call zones and for 6 four-hour intervals. In other words, different arrival rates are defined for each zone and each four-hour interval, rather than a homogeneous process characterized by a single average arrival rate. For the optimization model, mean daily demand at depot node j was estimated as

$$f_j(i) = P(j \mid i) \sum_{t=1}^{6} \lambda''_{it}$$

where j and i are depot and call nodes respectively, and λ''_{it} is the tth arrival rates for call node s. Historical and simulated average daily demand for EMS service system wide were both about 200 calls per day. Since there were 16 medic-units (depots) in the historical system, this average was used to define a maximum workload for any medic-unit. Perfectly balanced utilization would be achieved at about 12.5 calls per day per unit. A total of 28 current and potential medic-unit locations or home depots had been defined in the original study and were retained. A prescriptive model can be constructed by optimizing the figure of merit as defined by Equation 3.3, including equity among call zones.

Let $\Gamma(W, p) = \{j \mid y_j = 1\}$ be the siting result from the optimization of the location model with binary location-variable y_j, maximum workload for a medic-unit W, and p medic-units relocated. Solutions were generated for workloads $W = \infty$, and $W = 18$, 16, and 14. When the least constrained optimization was solved ($W = \infty$), it was found that the maximum number of units relocated was 6; thus the **relocation model** was solved for $p = 1, \ldots, 6$. Example formulations of such an optimization model can be found in the "Facility Location" and "Measuring Spatial Separation" chapters of Chan (2005). Illustrative software is also included in the enclosed CD/DVD under the SPACEFIL folder. Configuration solutions $\Gamma(w, p)$ to the various optimizations were obtained and, where possible, system performance measures for the configurations were compared using statistical inference. The optimization measures are deterministic and do not lend themselves to statistical comparison. All differences must be assumed to be significant since the average is all that is available. Simulation, however, also provides sample variance so the *t*-test[3] could be used to compare mean response times, $M(W, p)$, which were found by simulating the optimization solutions, $\Gamma(W, p)$. Here, *t*-values were calculated to compare $M(W, p)$ among themselves and with the base case.

The *t*-statistics indicate that the optimization solutions produced the desired effect of mean response time reduction in the simulated system. At about the 15 percent significance level, solutions to reduce mean response time were found by the optimization model. However, the simulated $W = \infty$ solutions, when compared to the base case, do not result in statistically significant mean response time changes, even at the 20 percent significance level. This result is quite different from the optimization results, in which up to a 0.14 minute mean response time reduction was achieved. Furthermore, if the significance level was reduced to 30 percent, it is possible that the solution, $\Gamma(\infty, p)$, for $p = 3$ and 6 results in increases in mean response time (t-values of 1.21 and 1.16 respectively). Finally, if significance levels were restricted to at least 10 percent, the t-values indicate that configuration solutions from the $W = \infty$ solution to relocate 3 to 6 units produced simulated mean response times significantly higher than the corresponding location solutions with maximum workload set at 16 and 18 respectively. These differences between simulation and optimization models can possibly be attributed to a homogeneous versus non-homogeneous assumption.

With the exception of these aberrations, optimal solution to the location of EMS has been verified against simulation results overall. Simulation was shown to be important and necessary for designing and verifying location models. Without the use of simulation, the effects projected from solutions produced by deterministic optimization models may be erroneous or not statistically significant. In the following section and the next chapter, we will describe in more detail the basic philosophy behind optimization models, so that the reader can better appreciate the statements made above. It should also be noted that this example illustrates that simulation has its proper place in spatial-temporal analysis. It can supplement stochastic process when the problem is too big to be solved analytically. It is also a good verification tool when real world data are not available for model validation.

With the advantage of several years, a similar verification study was performed by Repede and Bernardo (1994). An optimization model that maximizes the expected demand coverage was constructed. Demand variations over time or non-homogeneity was explicitly modeled in both the optimization and simulation models in the study. The siting and fleet size (or the number of ambulances) decisions obtained from the optimization model were input to a simulation model that detailed the performance of the location and fleet size decisions. The computational cycle between the two models formed a decision support system for the city of Louisville, Kentucky, with the optimization location model prescribing the siting and the simulation model evaluating the siting decision. In this case, validation data were available to gauge the quality of both the conventional static optimization model and the dynamic decision-support systems model proposed here. The decision-support system, which explicitly recognized non-homogeneity, was found to yield better agreement with field data. Better still, the decision-support system arrived at improved location decisions, corresponding to a 13-percent increase in coverage and 36 percent decrease in response time. This study again reinforces the role of stochastic process—and simulation in particular—in location analysis.

VI. INVENTORY CONTROL USING MARGINAL ANALYSIS

In facility location, siting of warehouses is often of interest. The intent is to place a warehouse so that the transportation cost of resupplying its inventory is lowest, and so is the delivery cost to the stores or customers. Simulation and stochastic process can be used in inventory analysis, wherein the decision to reorder can be reached systematically to avoid stockout and excessive storage costs. A common approach is to employ **marginal analysis,** which shows how a descriptive tool can be used for optimization in special cases (Lapin 1975, Winston 1994). When the demand for the inventory in the warehouse is uncertain and its lifetime is limited, we have an example of the newsboy problem.[4] Similar to selling newspapers at the newsstand, the objective of the newsboy problem is to decide how many items should be ordered at the beginning of each inventory cycle. The **uncertain demand** during the period $d\xi$ expresses the number of items that customers will require during this period. Two types of outcomes may occur. If demand is no larger than the order quantity (z), sales will equal the quantity demanded $d\xi$. If demand is greater than the initial order, sales will be limited to the order quantity z.

Three cost elements are considered: C is the unit cost of surplus inventory, and c is the unit opportunity cost due to shortage. Let $F(\xi)$ be the cumulative demand-function where ξ is the random variable for demand. To obtain the expected cost, we will assume that the probabilities for possible levels of demand $dF(\xi)$ is known. For an order quantity z and when demand is $d\xi$, the total cost is

$$\begin{array}{ll} C(z - d\xi) & \text{if } d\xi \leq z \\ c(d\xi - z) & \text{if } d\xi > z \end{array} \tag{3.4}$$

The expected total cost is $\varphi = CP(z \geq d\xi) + cP(z < d\xi)$. Such a cost function is plotted in Figure 3.9. Equation 3.4 says that if demand is less than the order quantity, we have overstocked at a per-unit cost of C, and if demand is higher than the order quantity, we have understocked at a per-unit cost of c. Should we order one additional unit dz, the surplus cost will be increased by C, but the stockout cost will be reduced by c (or a $-c$ change). This marginal cost can be represented by the difference between overstocking cost and the reduction in stockout cost:

$$\varphi(z + dz) - \varphi(z) = CP(z \geq d\xi) - cP(z < d\xi)$$

To order the optimal quantity, we order until the marginal cost is equal to zero, assuming that we have the common situation of a convex cost function as shown in Figure 3.9. In other words, starting out with having economy of scale, we order until it starts to change over to dis-economy of scale:

$$CP(d\xi \leq z) - c(1 - P[d\xi \leq z]) \geq 0$$

The optimal order quantity z^* is the smallest level z such that

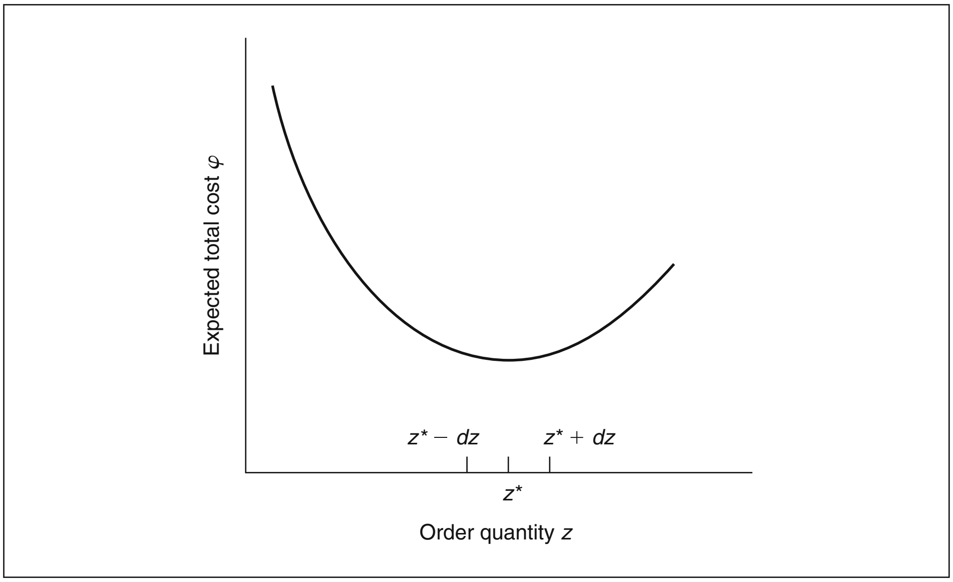
$$CP(d\xi \leq z) - c(1 - P[d\xi \leq z]) = (C + c)\, P(d\xi \leq z) - c \geq 0 \tag{3.5}$$

or

$$P(d\xi \leq z) = F(z) \geq \frac{c}{C + c} \tag{3.6}$$

This tells us that we need to calculate only the above ratio using the unit costs given for the problem and establish the cumulative probability for demand. The smallest demand z^* with a cumulative probability that exceeds this ratio is the order quantity that minimizes total expected cost.

Figure 3.9 A CONVEX EXPECTED TOTAL-COST FUNCTION



Example

Instead of direct substitution of numbers, we illustrate in this example how such a result as shown in Equation 3.6 can be useful in parametric analysis (Faneuff, Sterle, and Chan 1992). It also shows that marginal analysis has its analytic properties, thus surpassing simulation in the modeling process in limiting cases. Let demand be a uniform probability density function $f(\xi) = 0.002$. The cumulative function is then correspondingly $F(\xi) = 0.002\xi$. Suppose the current inventory stands at 100 units, and the warehouse has a storage capacity of 500. Also the expected total cost $\varphi(z)$ for an inventory cycle is parametric on Ω and ρ: $2600 - (8 + \Omega + \rho)z$. The unit costs of over- and under stocking (C and c) can be determined as a function of Ω and ρ by solving this equation:

$$2600 - (8 + \Omega + \rho)z = \int_{100+z}^{500} c(0.002d\xi) + \int_{0}^{100+z} C(0.002d\xi) \qquad (3.7)$$

where $dF(\xi) = 0.002d\xi$ as mentioned. The right-hand side of the above equation boils down to $(0.8c + 0.2C) + (0.002C - 0.002c)z$, which as a function of z consists of the intercept $(0.8c + 0.2C)$ and the slope $(0.002C - 0.002c)$. By equating them to the corresponding intercept and slope on the left-hand side of Equation 3.7 respectively, we have two equations and two unknowns C and c:

$$0.8c + 0.2C = 2600$$
$$0.002(C{-}c) = -8{-}\Omega{-}\rho$$

This result is $C = -400(\Omega + \rho) - 600$ and $c = 100(\Omega + \rho) + 3400$.

According to Equation 3.5, the marginal cost is set at zero, $\frac{d\varphi(z)}{dz} = 0$, for an optimal order quantity z^*, or $\frac{d\varphi(z)}{dz} = (C + c)\,F(z) - c = 0$. Substituting the values for C, c and $F(z)$ from above,

$$\frac{d\varphi(z)}{dz} = (5.6 - 0.6\Omega - 0.6\rho)z - 100(\Omega + \rho) - 3400 = 0 \tag{3.8}$$

which determines the optimal order quantity in terms of the parameters Ω and ρ

$$z^* = \frac{100(\Omega + \rho) - 3400}{5.6 - 0.6\Omega - 0.6\rho} \tag{3.9}$$

In locating a warehouse, order quantity is specified for each location. This is illustrated in the "Simultaneous Location-Routing" chapter of Chan (2005). Here Ω is the parameter to account for the given warehouse capacity and ρ for each delivery vehicle capacity—capacity that is needed to deliver supplies to a demand location. ■

VII. BAYESIAN ANALYSIS

Similar to inventory control, many facility location and land use analyses require a rule to help make sound decisions. A descriptive tool to accomplish this is **Bayesian analysis,** which includes the latest information on top of past knowledge in formulating the decision rule. We will describe this method via an example of constructing nuclear power plants on a proposed site considering safety and other environmental conditions. Based on site-specific environmental studies, we can summarize the results in terms of two states of nature: (a) geotechnical condition ideal and (b) condition marginal. Historical record of site-specific studies have shown that one out of 10 sites in the study area shows up as suitable, or P(ideal) = 0.1. This means that P(marginal) = 0.9, or nine out of 10 sites are not suitable. Now the decision maker faces two actions to consider for the construction project: (a) to build or (b) not to build at the proposed site. A payoff matrix can be written for the savings in millions of dollars for each nuclear power plant corresponding to the various decisions and states of nature:

	ideal	marginal
build	1.3	−0.2
not-to-build	0	0

This payoff matrix says that if we decide to build at the site, and the site turns out to be ideal, \$1.3 million will be saved. On the other hand, if the site turns out to be marginal, remedial engineering will cost an additional \$200,000. Obviously, a decision not to build does not incur any savings or cost. The expected return for each decision can now be computed:

$$\begin{aligned} \textit{build}&: \ 0.1(1.3) + 0.9(-0.2) = -0.05 \\ \textit{not-to-build}&: \ 0.1(0) + 0.9(0) = 0 \end{aligned} \tag{3.10}$$

Based on these numbers, the not-to-build decision should be the course of action since it is less costly.

A. Bayesian Update

Instead of making the final decision based on historical records, it is decided that additional information is to be gathered in order to have better knowledge about the site. The immediate decision then involves having additional borings drilled at $100,000 each or no additional work at (naturally) no cost. The boring may result in a positive statement about the suitability of the site or a negative statement. From experience, the accuracy of boring tests are as follows:

$P(positive \mid ideal) = 0.7$ [or $P(negative \mid ideal) = 0.3$]
$P(positive \mid marginal) = 0.2$ [or $P(negative \mid marginal) = 0.8$]

This says that the chance that a boring will tell the true story when the site is suitable is 70 percent and the chance that it will tell the wrong story is 30 percent. Likewise, the chance of telling the truth when the site is not suitable is 80 percent and telling a lie 20 percent.

The reliability of the boring test can be represented by graphical means in terms of events and sample elements. The following diagram summarizes the possible outcomes:

	ideal	*marginal*
positive	*iiiiiii*	*mm*
negative	*iii*	*mmmmmmmm*

$m = 9\,i$

This indicates that the chance that the test will turn out to be positive given the site is ideal (*i*) is seven out of 10, and that the test will turn out to be negative is three out of 10. Similarly, the chance that the test will show up negative when the site is marginal (*m*) is eight chances out of 10, while the chance that it will show up positive is two out of 10 times. For every ideal site, there are nine marginal sites. This event diagram is a convenient way of keeping track of sample elements. With this diagrammatic representation, the **Bayes' rule** can be easily explained and summarized by the following equation

$$P(ideal \mid positive) = \frac{P(ideal\ and\ positive)}{P(positive)} \tag{3.11}$$

This equation shows how to compute the chance that the site is ideal given a positive result of a boring test, when all the information is available on the right-hand side of the equation. Assuming the probability of a site being ideal given that a positive boring test is not available, such a calculation is necessary to arrive at a build or not-to-build decision. Intuitively, if the boring test is positive, we tend to infer the site is ideal, which can lead toward a build decision. Obviously, this inference can be made only under the following condition: The probability of the site being ideal given a positive test is high. Thus the decision is made easy by evaluating the above equation or assembling all the information on the right-hand side of the equation.

The assemblage of this general set of information is expedited by the application of Bayes' rule, which can be represented conveniently by Figure 3.10, a companion to the event diagram above. Assisted by the figure, these calculations can be made by further breaking down the right-hand side of Equation 3.11 in terms of the given information:

$$\begin{aligned} P(\textit{ideal} \text{ and } \textit{positive}) &= P(\textit{ideal})P(\textit{positive} \mid \textit{ideal}) \\ &= (0.1)\ (0.7) \\ &= 0.07 \end{aligned}$$

$$\begin{aligned} P(\textit{positive}) &= P(\textit{ideal} \text{ and } \textit{positive}) + P(\textit{marginal} \text{ and } \textit{positive}) \\ &= P(\textit{ideal})\ P(\textit{positive} \mid \textit{ideal}) + P(\textit{marginal})\ P(\textit{positive} \mid \textit{marginal}) \\ &= (0.1)(0.7) + (0.9)(0.2) \\ &= 0.25 \end{aligned}$$

which lead toward $P(\textit{ideal} \mid \textit{positive}) = (0.07)/(0.25) = 0.28$. These calculations can be confirmed by the event diagram above by counting the sample elements row-wise and then column-wise for the event of interest. Thus the probability of being ideal *and* positive is

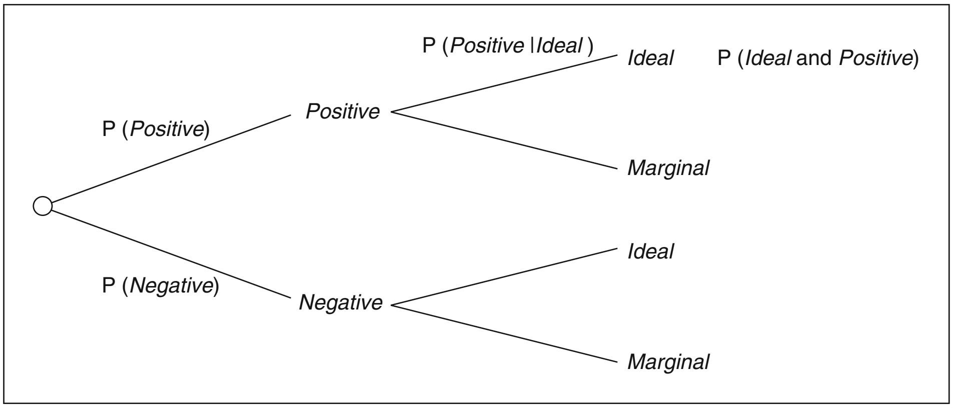
$$7i/(7i + 3i + 2m + 8m) = 7i/(7i + 3i + 2[9i] + 8[9i]) = 0.07 \textit{ and so on}$$

B. Bayesian Decisions

Given the result of an additional boring is positive, the new payoff matrix, after inclusion of the boring cost in dollars, is

	ideal	marginal
build	1.2	−0.3
not-to-build	−0.1	−0.1

Figure 3.10 GRAPHIC REPRESENTATION OF BAYES' RULE



The expected return (in dollars) of the decision to build can be computed as

$$\begin{aligned}&P(ideal \mid positive)(1.2) + \mathrm{P}(marginal \mid positive)(-0.3)\\&= (0.28)(1.2) + (0.72)(-0.3)\\&= 0.12.\end{aligned}$$

Since the expected return of the not-to-build decision is −0.1, the decision is clearly to build if boring results are positive. Next is to decide between build or not-to-build if the test is negative. Again, employing Bayes' rule:

$$P(ideal \mid negative) = \frac{P(ideal \text{ and } negative)}{P(negative)}$$

and following the same calculations as in the case of a positive boring result, it can be shown that this conditional probability is evaluated at 0.04. The expected return of building a plant is −0.24, and the expected return of the not-to-build decision is −0.1. Thus the decision is clearly not-to-build if the test result is negative.

The next logical question is: Should tests be conducted? In other words, can we arrive at the same decision without the extra expense and trouble of boring tests? To answer this question, we calculate the expected return if testing is conducted:

$$\begin{aligned}&P(positive)(Payoff\ of\ a\ positive\ result) + P(negative)(Payoff\ of\ a\ negative\ result)\\&= (0.25)\{P(ideal \mid positive)\,[1.2] + P(marginal \mid positive)[-0.3]\} +\\&\quad (0.75)\{P(ideal \mid negative)\,[-0.1] + P(marginal \mid negative)[-0.1]\}\\&= (0.25)[0.28(1.2) + 0.72(-0.3)] + (0.75)[0.04(-0.1) + 0.96(-0.1)]\\&= -0.045.\end{aligned}$$

Likewise, we calculate the expected return if testing is not conducted, which is the same as the expected return of the not-to-build decision as computed in Equation 3.10:

$$\begin{aligned}&P(ideal)(payoff\ from\ an\ ideal\ condition) +\\&P(marginal)(payoff\ from\ a\ marginal\ condition)\\&= (0.1)(0) + (0.9)(0)\\&= 0\end{aligned}$$

Compared with incurring a cost of $100,000 to conduct a test, the decision is obviously not to test.

C. Decision Tree

The best way to review the entire problem is by way of a **decision tree,** which summarizes all the possible decisions and outcomes. Referring to the decision tree of Figure 3.11, these with the given data are displayed. (a) The payoff matrices are laid out in the *Payoff* column. (b) The cost of a test is 0.1 million dollars. This means the cost of constructing a power plant is (0.3 − 0.1) = 0.2 million dollars.

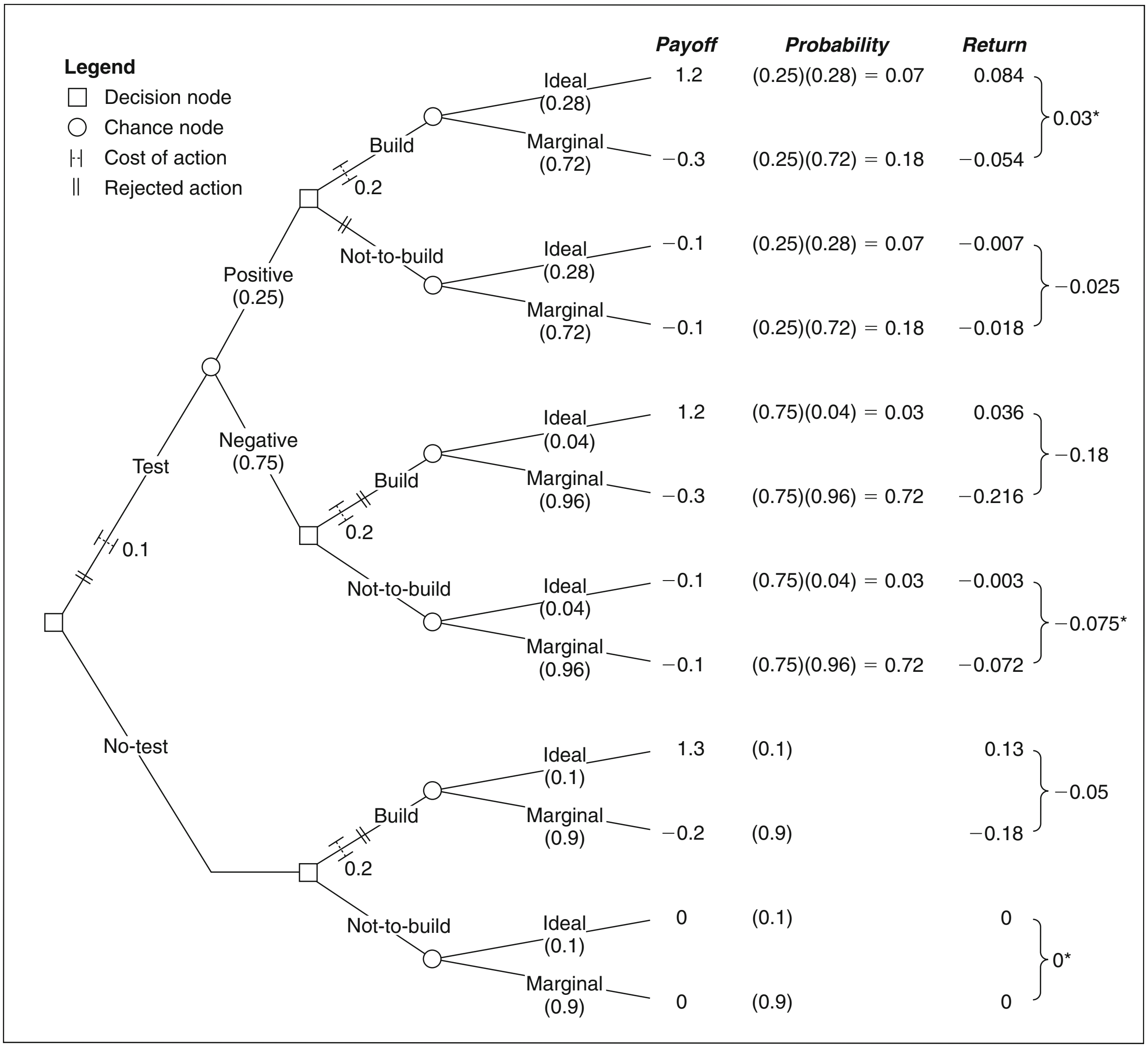
With the previous calculations summarized in the same figure, it can be verified that the expected return of building the plant, given testing is positive, can be easily computed from the information contained in Figure 3.11: $E(build \mid positive) = (\$0.03)/0.25 = \$0.12$, which says that the expected return given a boring test turns out to be positive is $120,000. Similarly, the expected return of a not-to-build de-cision given a test is positive is:

$$E(not\text{-}to\text{-}build \mid positive) = (-\$0.025)/0.25 = -\$0.1$$

which amounts to a cost of $100,000. In the same way,

$$E(build \mid negative) = (-\$0.18)/0.75 = -\$0.24.$$
$$E(no\text{-}build \mid negative) = (-\$.075)/0.75 = \$-0.1.$$

Figure 3.11 BAYESIAN DECISION TREE



Note: For the "Not-to-build" branch, it is not necessary to include the chance node.

Notice from the decision tree that it is meaningless to ask for the expected return of a build decision, $E(build)$, without the qualification of "given the test is positive," "given the test is negative," or "given there is no test." However, it is very meaningful to calculate the expected return of conducting a test. Thus following the logical paths in the decision tree, $E(test)$ can be easily evaluated: $E(test) = (\$0.03) + (-\$0.075) = -\$0.045$. Following a similar path on the other side of the decision tree, the expected return of the no-test decision, $E(no\text{-}test)$, is obviously 0 even without any computation.

To round out the discussion on Bayesian decision making, we like to ask the question: "How much is the boring worth in terms of the additional information it buys?" This is referred to as the expected value of sample information, and can be computed as

$$\begin{aligned}&(\textit{expected value of optimal decision with sample info}) - \\ &\qquad (\textit{expected value of optimal decision without sample info}) \\ &\qquad = \$[(-0.045) - (0)] \\ &\qquad = \$0.045\end{aligned}$$

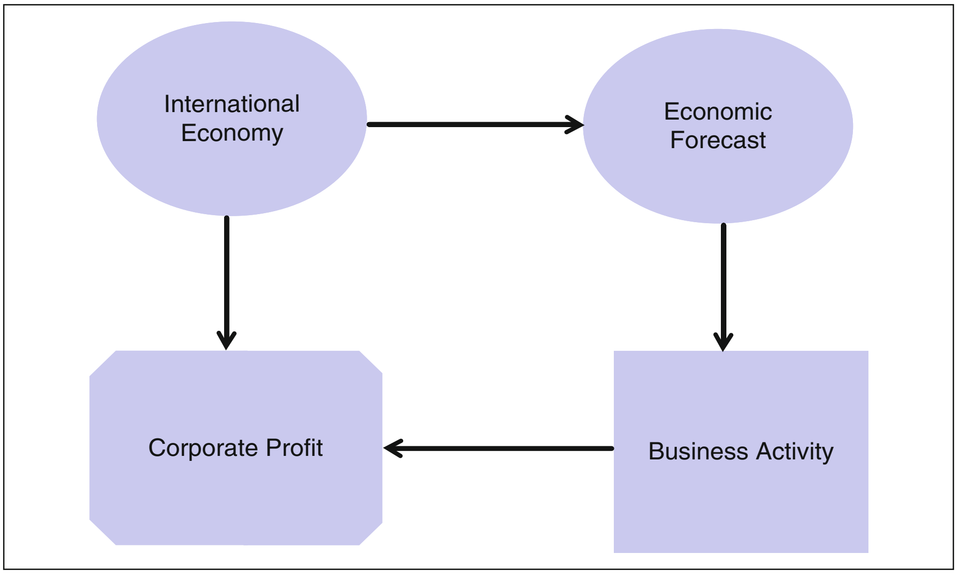
Thus, the decision maker should be willing to pay up to $\$(-45{,}000 - (-100{,}000)) = \$55{,}000$ for an additional boring test (sample information). (One can think of this calculation as adding back the $100,000 sampling cost into the decision tree, in order to fully account for the full worth of the sampling test.) Since the test actually costs $100,000, we conclude that the test is not worthwhile. Consequently, following the logical path to its logical conclusion on the decision tree, building the plant at the proposed site is undesirable.

The above case study is a simple way to introduce **Decision Analysis**. In this case study, a single metric, expression in dollars, is used throughout as the criterion for evaluation. It is obvious that this represents the simplest case when every measure can be quantified conveniently in terms of a cost expressed in dollars. This subject will be further discussed in Chapter 5, entitled "Multicriteria Decision Making," where multiple metrics will be used to choose between alternatives. This generalization is in many ways a logical progression, as pointed out by Tsoukias (2008), who argues eloquently that the decision-analysis procedure is simply a subset of the body of knowledge known as **Decision Aiding Methodology**.

D. Influence Diagram

As an alternative to a Decision Tree, an **influence diagram** provides a simple graphical representation of a decision problem (Clemens 1996; Wikipedia 2010). It is a generalization of a *Bayesian network*, in which not only probabilistic inference problems but also decision-making problems can be modeled and solved. By a decision-making problem, we mean choosing among alternatives following the maximum expected utility criterion. The elements of a decision problem—decisions to make, uncertain events, and the value of outcomes—show up in the influence diagram as differently-shaped nodes. These nodes are then linked with arrows in specific ways to show the relationships among them. Influence diagram is now adopted widely and becoming an alternative to a decision tree which typically suffers from exponential growth in number of branches with each variable modeled, as one can gather from even the simple problem above.

Figure 3.12 AN EXAMPLE INFLUENCE DIAGRAM



Let us illustrate with a simple influence diagram for making decision about business activities. Consider the simple influence diagram in Figure 3.12, representing a situation where a business is planning its transactions. There is one **decision node** (Business Activity), two **uncertainty nodes** (International Economy, Economic Forecast), and one **value node** (Corporate Profit).

There are two functional arcs that end in the "Corporate Profit" node, one conditional arc that ends in the "Economic Forecast" node, and one informational arc that ends in the "Business Activity" node. The functional arcs ending in Corporate Profit indicate that Corporate Profit is a function of the International Economy and Business Activity. In other words, Corporate Profit can be estimated if the business knows what the International Economy is like and what its choice of activity is. (Notice that in this strict relationship it does not value Economic Forecast directly in estimating Corporate Profit.)

The conditional arc ending in Economic Forecast indicates the business's belief that Economic Forecast and the International Economy can be dependent. The informational arc ending in Business Activity indicates, however, that the business will only have knowledge of the Economic Forecast, not the actual International Economy, when making its choice. Stated differently, the actual economic condition will be known after it makes its choice (not before). Only the forecast is all it can count on at this stage. It also follows, semantically, that Business Activity is independent of (or irrelevant to) International Economy, given the Economic Forecast is all that is available for decision-making.

The above example further illustrates the power of an influence diagram in representing an extremely important concept in decision analysis known as value of information. Consider the following three scenarios:

- □ Scenario 1: The business could make its Business Activity decision while knowing what International Economy will be like. This corresponds to adding an extra informational arc from International Economy to Business Activity in the above influence diagram.

□ SCENARIO 2: This is represented by the original influence diagram as shown in Figure 3.12.

□ SCENARIO 3: The business makes its decision without even knowing the Economic Forecast. This corresponds to removing the informational arc from Economic Forecast to Business Activity in Figure 3.12.

Scenario 1 is the best possible scenario for this decision situation since there is no longer any uncertainty on what the business wants to know (about the International Economy) when making its decision. Scenario 3, on the other hand, is the worst possible decision situation since the business needs to make its decision without any knowledge (not even an Economic Forecast) on the actual International Economy. The decision-maker is usually better off (definitely no worse off) to move from scenario 3 to scenario 2 through the acquisition of new information. The most it should be willing to pay for such a move is called value of information on Economic Forecast, which is essentially the value of imperfect information on the International Economy.

Likewise, it is the best for the business to move from scenario 3 to scenario 1. The most it should be willing to pay for such a move is called value of perfect information on the International Economy.

E. Bayesian Classifier

The reader may be so convinced by the nuclear power plant example above that it is possible to formalize a rule to classify go versus no-go decisions. But what about situations where there are more than two decisions—for instance, "go," "no-go," and "wait"—and is there a more compact way to characterize the decision in such situations? Here we have a general classification problem, deciding which logical decision among K decisions (where $K > 2$) we should commit ourselves to given certain payoffs and probabilities of outcomes. A classification exists that is optimal in terms of expected payoffs, and also yields the lowest expected probability of committing classification errors (Gonzalez and Woods 1992).

Instead of a single attribute (such as dollars in the above siting example), let a decision be made based on a vector of attributes $\mathbf{x} = (x_1, x_2, \ldots)$ where x_1 may be cost, x_2 may be risk, and so on. The probability that a particular vector of attributes $\mathbf{x}$ logically belong to class G_i is denoted by $P(G_i \mid \mathbf{x})$. If a classifier decides that $\mathbf{x}$ logically belonged to G_j when it actually belonged to G_i, it incurs a classification error, which is manifested in terms of a loss measure L_{ij} (Rue 1995). As attribute vector $\mathbf{x}$ may belong to any of K classes under consideration, the average loss incurred in assigning $\mathbf{x}$ to class G_j is

$$L_j(\mathbf{X}) = \sum_{k=1}^{K} L_{kj} P(G_k \mid \mathbf{x})$$

Using Bayes' rule, or $P(A \mid B) = [P(A)P(B \mid A)]/P(B)$, the above equation can be re-written as

$$L_j(\mathbf{x}) = \frac{1}{P(\mathbf{x})} \sum_{k=1}^{K} L_{kj} P(\mathbf{x} \mid G_k) P(G_k) \qquad (3.12)$$

where $P(\mathbf{x} \mid G_k)$ is the probability that the attribute or feature vector really comes from class G_k and $P(G_k)$ is the probability of occurrence of class G_k. Since $1/P(\mathbf{x})$ is common to all the loss measures $L_j(\mathbf{x})$, $j = 1, 2, \ldots, K$; it can be dropped from Equation 3.12 without affecting the relative order of these functions from the smallest to the largest value. The expression for the average loss then reduces to

$$L_j(\mathbf{x}) = \sum_{k=1}^{K} L_{kj} P(\mathbf{x} \mid G_k) P(G_k) \tag{3.13}$$

The classifier has K possible classes to choose from for any given feature vector $\boldsymbol{x}$. It computes $L_1(\mathbf{x}), L_2(\mathbf{x}), \ldots, L_K(\mathbf{x})$ and assigns the feature vector to the class with the smallest loss. In many decision problems, the loss for a correct decision is zero, and it has the same non-zero value (for example, 1) for any incorrect decision. Under these conditions, the loss function becomes

$$L_{ij} = 1 - z_{ij} \tag{3.14}$$

where the indicator variable $z_{ij} = 1$ if the vector has been properly classified ($i = j$). On the other hand $z_{ij} = 0$ if it is improperly classified ($i \neq j$). Equation 3.14 indicates a loss of unity for incorrect decisions and zero loss for correct decisions (as indicated by the indicator $z_{ii} = 1$ or in vector notation $\mathbf{Z}_i = (z_{ii}, z_{ij})^T = (1, 0)^T$). Substituting this equation into Equation 3.13 yields the following expressions. Please note that for the first term of the summation expansion, it simply suggests that P(A | B)P(B) + P(A | ~B)P(~B) = P(A), where ~B is the complement of event B. For the second term of the summation expansion, all $z_{kj} = 0$ except when $k = j$ (or $z_{ij} = 1$).

$$\begin{aligned} L_j(\mathbf{x}) &= \sum_{k=1}^{K} (1 - z_{kj}) P(\mathbf{x} \mid \mathbf{z}_k) P(\mathbf{z}_k) \\ &= \sum_{k=1}^{K} \left[P(\mathbf{x} \mid \mathbf{z}_k) \mathrm{P}(\mathbf{z}_k) - \mathbf{z}_{kj} P(\mathbf{x} \mid \mathbf{z}_k) P(\mathbf{z}_k) \right] \\ &= P(\mathbf{x}) - P(\mathbf{x} \mid \mathbf{z}_j) P(\mathbf{z}_j). \end{aligned} \tag{3.15}$$

The Bayes' classifier then assigns a feature vector $\mathbf{x}$ to class G_i if $L_i(\mathbf{x}) < L_j(\mathbf{x})$, or

$$P(\mathbf{x}) - P(\mathbf{x} \mid \mathbf{z}_i) P(\mathbf{z}_i) < P(\mathbf{x}) - P(\mathbf{x} \mid \mathbf{z}_j P(\mathbf{z}_j) \tag{3.16}$$

This is equivalent to

$$P(\mathbf{x} \mid \mathbf{z}_i) P(\mathbf{z}_i) > P(\mathbf{x} \mid \mathbf{z}_j) P(\mathbf{z}_j) \qquad j = 1, 2, \ldots, K;\ j \neq i \tag{3.17}$$

Thus we can see that the Bayesian classifier for 0-1 loss functions is nothing more than implementation of decision function of the form

$$L_j'(\mathbf{x}) = P(\mathbf{x} \mid \mathbf{z}_j) P(\mathbf{z}_j) \qquad j = 1, 2, \ldots, K \tag{3.18}$$

where a feature vector $\mathbf{x}$ is assigned to class G_i if $L_i'(\mathbf{x}) > L_j'(\mathbf{x})$ for all $j \neq i$.

As an example, consider a scalar attribute x involving two classifications ($K = 2$) governed by Gaussian **probability-density functions (PDFs)**, with means μ_1 and μ_2 and standard deviations σ_1 and σ_2 respectively. From Equation 3.18, the decision function has the form

$$\begin{aligned} L'_j(x) &= P(x \mid \mathbf{z}_j)P(\mathbf{z}_j) \\ &= \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left[-\frac{(x-\mu_j)^2}{2\sigma_j^2} \right] P(z_j) \qquad j = 1, 2 \end{aligned} \tag{3.19}$$

Figure 3.13 shows a plot of the PDF for the two classes. The boundary between the two classes is a single point, x_0, such that $L'_1(x_0) = L'_2(x_0)$. If the two classes are equally likely to occur, $P(\mathbf{z}_1) = P(\mathbf{z}_2) = 1/2$, and the decision boundary is the value of x_0 for which $P(x_0 \mid \mathbf{z}_1) = P(x_0 \mid \mathbf{z}_2)$. This point is the intersection of the two PDFs, as shown in Figure 3.13. When $\mu_1 = 0$, $\mu_2 = 1$, and $\sigma_1 = \sigma_2 = \sigma$, for example, $x_0 = 1/2$. Any feature attribute to the right of $x_0 = 1/2$ is classified as belonging to class G_1. Similarly, any feature attribute to the left of $x_0 = 1/2$ is classified as belonging to class G_2. For computational ease, logarithm is often applied toward the decision function:

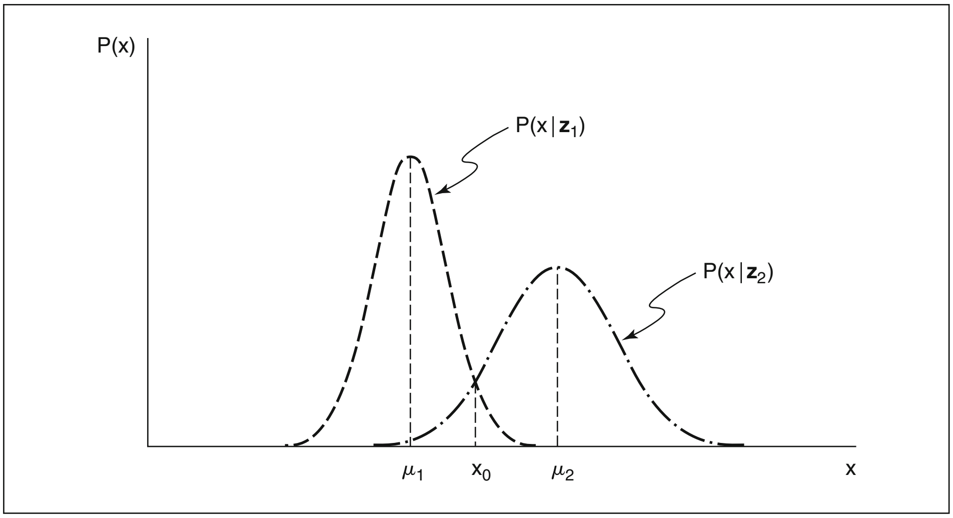
$$\begin{aligned} L''_j &= \log L'_j \\ &= \log [P(\mathbf{x} \mid \mathbf{z}_j)P(\mathbf{Z}_j)] \\ &= \log P(\mathbf{x} \mid \mathbf{z}_j) + \log P(\mathbf{z}_j) \end{aligned} \tag{3.20}$$

In the case of the scalar Gaussian PDF above, this simplifies to

$$\log P(\mathbf{z}_j) - \log \sigma_j - \frac{(x-\mu_j)^2}{\sigma_j^2} \tag{3.21}$$

after leaving out the common constant term such as $-1/2(\log 2\pi)$.

Figure 3.13 DEFINING A DECISION BOUNDARY



SOURCE: Gonzalez and Woods (1992). Reprinted with permission.

Now let us compare and contrast two classification possibilities by taking the ratio

$$\log\left[\frac{P(z_{1j}=1\mid \mathbf{x},\mathbf{z}_j)}{P(z_{2j}=1\mid \mathbf{x},\mathbf{z}_j)}\right].$$

For known parameters μ_1, μ_2, and σ_2, we have from Bayes' theorem:

$$\begin{aligned}P(A\mid BC) &= P(ABC)/P(BC) = P(ABC)/[P(B\mid C)P(C)] = \\ &P(A\mid C)P(C)P(B\mid AC)/[P(BC)P(C)] = P(A\mid C)/[P(B\mid C)/P(B\mid AC)]\end{aligned}$$

that

$$\begin{aligned}\log\left[\frac{P(z_{1j}=1\mid \mathbf{x},\mathbf{z}_j)}{P(z_{2j}=1\mid \mathbf{x},\mathbf{z}_j)}\right] &= \log\left[\frac{P(z_{1j}=1\mid \mathbf{z}_j)/(P(\mathbf{x}\mid \mathbf{z}_j)/P(\mathbf{x}\mid z_{1j}=1,\mathbf{z}_j))}{P(z_{2j}=1\mid \mathbf{z}_j)/(P(\mathbf{x}\mid \mathbf{z}_j)/P(\mathbf{x}\mid z_{2j}=1,\mathbf{z}_j))}\right] \\ &= \log\left[\frac{\exp(\beta T_{1j})}{\exp(\beta T_{2j})}\right] + \frac{1}{2}\left[-\frac{(x_j-0)^2}{\sigma^2} + \frac{(x_j-1)^2}{\sigma^2}\right] \qquad (3.22) \\ &= -\frac{\left(x_j-\frac{1}{2}\right)}{\sigma^2} + \beta(T_{1j}-T_{2j}).\end{aligned}$$

Notice that

$$\begin{aligned}\frac{P(\mathbf{x}\mid \mathbf{z}_j)/P(\mathbf{x}\mid z_{1j}=1,\mathbf{z}_j)}{P(\mathbf{x}\mid \mathbf{z}_j)/P(\mathbf{x}\mid z_{2j}=1,\mathbf{z}_j)} &= \frac{P(\mathbf{x}\mid \mathbf{z}_j)P(\mathbf{x}\mid z_{2j}=1,\mathbf{z}_j)}{P(\mathbf{x}\mid \mathbf{z}_j)P(\mathbf{x}\mid z_{1j}=1,\mathbf{z}_j)} \\ &= \left[\frac{(\exp\,[-(x_j-0)^2/2\sigma^2])}{(\exp\,[-(x_j-1)^2/2\sigma^2])}\right] \qquad (3.23)\end{aligned}$$

considering

$$P(\mathbf{x}\mid z_{ij}=1,\mathbf{z}_{\mathrm{j}}) = P(\mathbf{x}\mid \mathbf{z}_{\mathrm{j}}) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left[\frac{(x-\mu_j)^2}{-2\sigma^2}\right].$$

This shows that when $x_j = ½$, we have $P(\mathbf{x}\mid z_j)$ in both the numerator and denominator, corresponding to the joint probability of being in either group $j = 1$ or $j = 2$. On the other hand, when $x_j > ½$, the probability of being in group $j = 2$ is enhanced, and when $x_j < ½$, the probability of being in group $j = 1$ is enhanced. For those interested in a numerical illustration beyond the power plant example, please refer to the "Spectral versus Spatial Pattern Recognition" section in Chapter 6.

VIII. ECONOMETRIC APPROACH

The above described model building philosophy can be visualized as an approach wherein the casual sequence of events are chained together in a manner reflecting the process in real life, hence the terms decision analysis, simulation, and probabilistic models. Thus in decision analysis, we update a prior probability using sample information and based on the updated information define a decision boundary to classify a multi-attribute observation. In a stochastic model, one schedules service vehicles in response to time varying demands. Finally in simulation, we replicate the sequence of events in which a previous event leads toward a subsequent event. A parallel approach, to be discussed here, does not claim to have an explicit understanding of the causal chain. Rather, it examines a set of historic data and tries to postulate a structural relationship that explains the observed data. If history repeats itself, or if the structural relationship prevails, one can forecast the future. Such an approach, as applied in facility location and land use, is termed the **econometric method.** Its components are explained below.

A. Arrow Diagram and Path Analysis

The first step in an econometric approach is the construction of arrow diagrams. An **arrow diagram** is a graphic aid to postulate the relationships between a number of factors. These may be a primary, secondary, or tertiary order correlation. Note that an arrow diagram shows structural relationships and unlike its analogue, the logic flow chart, no casual pattern is implied. Thus in Figure 3.14, a relationship is postulated between the population in the base-year and the forecast-year (with an arrow pointing from base- to forecast-year), which expresses correlation and not causation. The distinction is really apparent if one compares it to the logic flow diagram that traces service activities to basic employment. While one may suspect that the population in the forecast period would continue to be large if population in the base period is large (a correlation), it is not the same as the more close ties between basic and nonbasic activities (a causation). Primary, secondary, and tertiary ordering (or relationship) between two factors is defined to reflect a decreasing degree of correlation. In the example shown in Figure 3.14, the most important correlation, according to the postulation of the model builder, exists between the base-year and the future-year population. The least dominant correlation, on the other hand, is that between base-year population and forecast employment.

Path analysis is a refinement of the arrow diagram technique. While the arrow diagram quantifies the correlation between two factors such as employment and population, path analysis defines the relationship more precisely by confirming the arrow from employment to population is in fact correct or vice versa, as shown

Figure 3.14 ARROW DIAGRAM FOR ECONOMIC-BASE EXAMPLE

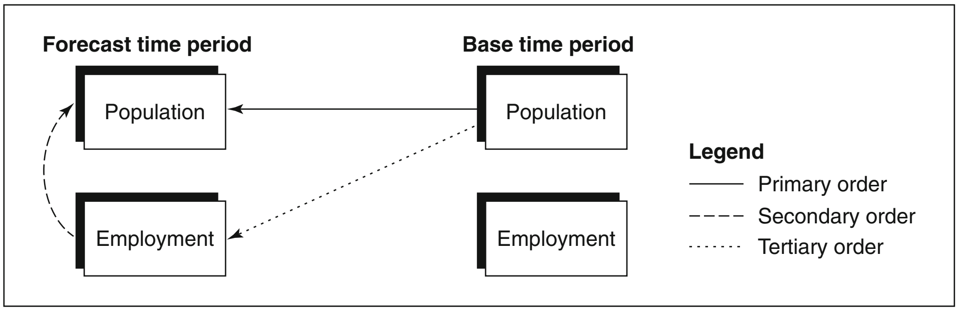
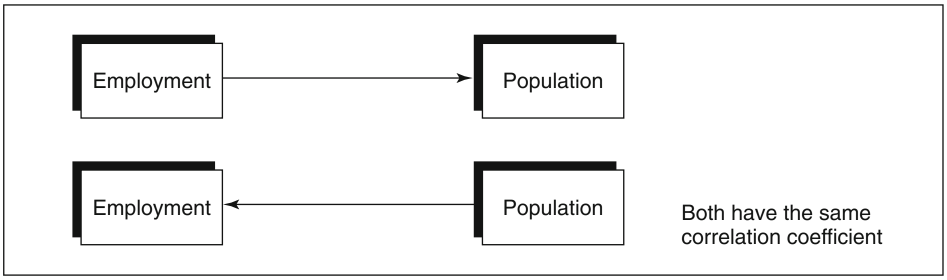


Figure 3.15 PATH ANALYSIS FOR THE ECONOMIC-BASE EXAMPLE

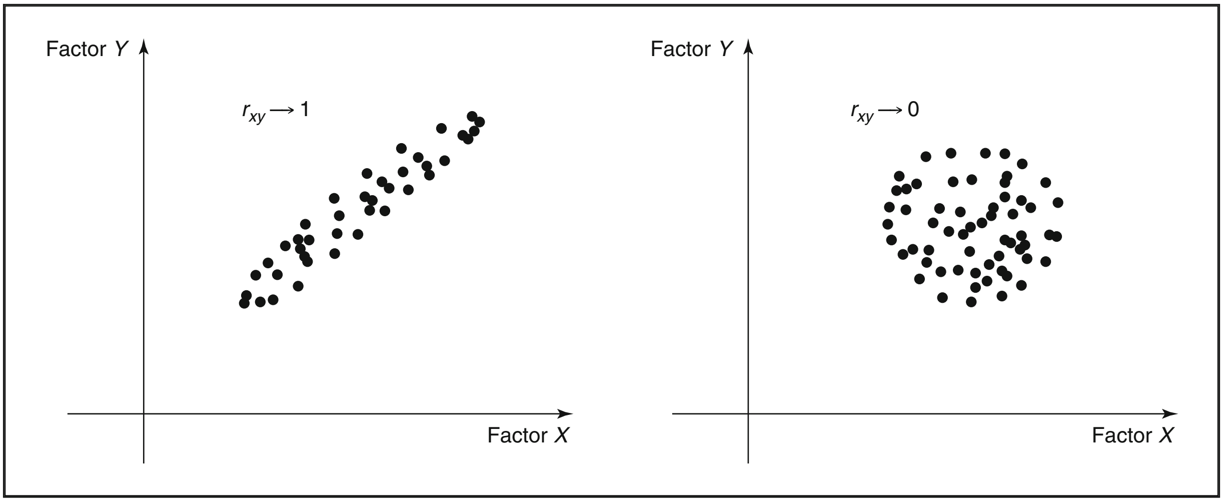


in Figure 3.15. Putting the discussion in more familiar experiences, path analysis addresses the chicken-and-egg phenomenon, where the initiating factor is to be identified. Citing a popular example, we may have found a high correlation between the number of medical doctors and the number of sick people in an infected community. A distinction is to be made between the sequence of events as to whether the infirm triggered the arrival of the doctors or the doctors caused the epidemic. The answer is quite obvious in this example. It is, however, less so in many other circumstances. Back to the economic base example we have been using, it is not entirely clear which is the initiating event: employment or population. As illustrated above in the economic base example, employment and population serve as the initiator alternatively. Path analysis can be used to resolve these nebulous situations because it is a means to check internal consistency—thus pointing out contradictory structural ordering. We will come back to this a bit later.

B. Econometric Models

Thus far, only the qualitative relationship between factors has been discussed. To quantify this relationship, we need to place a numerical value between each pair of factors. This is termed the **correlation coefficient,** which assumes a value from zero through unity[5] (Figure 3.16). A value close to unity would denote a high

Figure 3.16 CORRELATION COEFFICIENTS



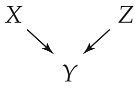
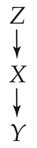
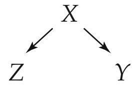
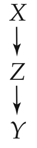
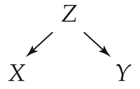
degree of association, while a value close to zero would indicate a lack of association between two factors. Once the correlation coefficient between population and employment (r_{PE}) is defined, mathematical expressions can be written for the arrow model we examine above: employment is proportional to population, or population is proportional to employment; where $r_{EP} = 1/r_{PE}$.

Once the correlation coefficients are defined, it is rather straightforward to recognize that the arrow diagram shown in Figure 3.14 can be quantified as the following set of equations:

$$\begin{aligned}(\textit{forecast pop}) &= a(\textit{forecast emp}) + b(\textit{base-yr pop}) \\ (\textit{forecast emp}) &= c(\textit{forecast pop}) + d(\textit{base-yr emp})\end{aligned} \tag{3.24}$$

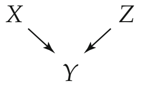
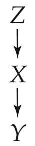
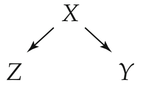
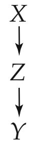
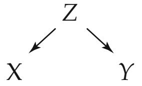
Here a, b, c and d are **calibration coefficients,** showing that future-year population is correlated both to base-year population and future-year employment. Now we are at a position to come back to the use of path analysis to validate a postulated set of relationships, which so far has been nothing but a hypothesis in the mind of the modeler. Application of path analysis to the 2-arrow models as shown in Table 3.4 results in the necessary conditions shown in Table 3.5. These tables illustrate two points. First, there are subsets of models where predicted relationships are mutually contradictory. Second, several models should, if valid, satisfy the same necessary conditions on the correlation coefficients.

Table 3.4 TYPOLOGY OF 2-ARROW MODELS

Model type	**Arrow diagram**	**Econometric equations**
Both X and Z independently affect Y.		$X = b_1$ $a_{21}X + Y + a_{23}Z = b_2$ $Z = b_3$
X, partially caused by Z, causes Y.		$X + a_{13}Z = b_1$ $a_{21}X + Y = b_2$ $Z = b_3$
The primary variable X causes both Z and Y.		$X = b_1$ $a_{21}X + Y = b_2$ $a_{31}X + Z = b_3$
The secondary variable Z intervenes between X and Y.		$X = b_1$ $Y + a_{23}Z = b_2$ $a_{31}X + Z = b_3$
The primary variable X and the supposedly dependently variable Y are correlated but not casually connected.		$X + a_{13}Z = b_1$ $Y + a_{23}Z = b_2$ $Z = b_3$

SOURCE: De Neufville and Stafford (1971). Reprinted with permission.

Table 3.5 RESULTS OF PATH ANALYSIS

Grouping of models	Arrow diagram	Path analysis prediction	Condition
Y in the middle		$r_{XZ} = 0$	$r_{XY} \neq 0$ $r_{YZ} \neq 0$
X in the middle	 	$r_{XY} = \frac{r_{YZ}}{r_{XZ}}$	$r_{XY} \neq 0$ $r_{XZ} \neq 0$
Z in the middle	 	$r_{XY} = r_{XZ}r_{YZ}$	$r_{XZ} \neq 0$ $r_{YZ} \neq 0$

SOURCE: De Neufville and Stafford (1971). Reprinted with permission.

The two tables constructed for 2-arrow models demonstrate that path analysis can be a useful means to discriminate between some models and can help the analyst reject some models that are obviously false. For example, in order for the "X-in-the-middle" (and the corresponding econometric models) to be valid, the correlation coefficients r_{XY} and r_{XZ} must be non-zero. If any one of these two correlation coefficients happens to be zero, the "X-in-the-middle" model is proven to be invalid and some of the corresponding pointing directions of the arrows may need to be reversed (as in the "Z-in-the-middle" models). The results also show that correlation coefficients are not a useful guide for the positive identification of the best model. Many models with contradictory implications may satisfy the same correlation requirements. Table 3.5 shows, for example, that one cannot distinguish statistically between the "Z-in-the-middle" models: $X \leftarrow Z \rightarrow Y$, $X \rightarrow Z \leftarrow Y$, $X \rightarrow Z \rightarrow Y$ and $X \leftarrow Z \leftarrow Y$. To distinguish between these possibilities, the modeler must rely upon his understanding of the situation being modeled.

IX. CALIBRATION

In the above discussions, we have presented two parallel methodologies to construct a descriptive model, one based on casual relationships while the other is founded upon correlation inferences. We call them, for convenience, simulation and econometric models respectively. In order to make either one of these models operational, a calibration process has to be carried out. This is to estimate the parameters of the model, such as the average number of dependents per employee (or the reciprocal of labor force participation rate)—an economic-base example we are familiar with—or the coefficients *a* and *f* in Table 3.1.

A. Ordinary Least Squares

The calibration procedure is expedited if one deals with a linear relationship, since there are more software packages available for linear than nonlinear relationships. Nonlinear relationships can often be converted to linear ones as shown in the example below—as we have demonstrated in Equations 3.20 and 3.21.

Nonlinear form: $W = aX^bY^cZ^d$
Log-linear form: $\log W = \log a + b \log X + c \log Y + d \log Z$

Once a log-linear equation is obtained, the linear statistical techniques can be applied when one treats the logarithm of a variable as the observations. When carrying out such a procedure, however, care must be exercised in observing the normal distribution assumptions on error terms in linear-regression calibration techniques, as explained in Appendix 2.

There are about five categories of goodness of fit statistical techniques, ranging from the less sophisticated to the more involved. The first to be discussed is the manual procedure. Here, the ratio between the two variables may be used to estimate the parameter. For example, to estimate the average size of the household, the total population in the study area is divided by the number of households. To estimate the labor force participation rate, the number of employees is divided by the total population and so on. The next technique is **ordinary least squares (OLS)**, where a graphical plot of the pair of variables of interest is used to determine an equation, from which the values of parameters can be obtained. For example, industrial development may be directly related to accessibility in a linear equation: *development* = *a*(*access*) + *b*. A plot such as Figure 3.16 will have development as the Y axis and accessibility as the *X* axis. Linear regression will yield the numerical values of *a* and *b*, as explained in Appendix 2.

Where there is more than one equation to be fitted, **indirect least squares** and **two-stage least squares** are the appropriate methods. For illustration purposes, let us examine the following simultaneous equation set, where *a*, *b*, and *c* are to be calibrated:

$$\begin{aligned}(\textit{forecast pop}) &= a(\textit{forecast emp}) + b(\textit{base-yr pop})\\ (\textit{forecast emp}) &= \qquad\qquad\qquad\quad + c(\textit{base-yr pop}).\end{aligned}$$

It is to be noted that the equation set is a realization of the second block of arrow diagrams in Table 3.4 and Figure 3.14. The special property of such a set of equations allows simplification to be made on the structural form. The second equation can be readily substituted into the first one, resulting in a reduced form: *forecast pop* = *d*(*base-yr pop*) + *e*, where the coefficients *d* and *e* can be calibrated using ordinary least squares techniques. The substitution in effect removes the coupling between the forecast employment variable that appears on the left-hand side of the second equation and the right-hand side of the first equation.

The name indirect least squares refers to the fact that through the reduced form straightforward regression can be performed on a single equation instead of the simultaneous set. In general, an exactly identified set of structural equations can be reduced to a number of uncoupled equations, which can then be calibrated independent of one another.

B. Two-Stage Least Squares

If an arrow diagram results in the following equations:

$$\begin{aligned}(\textit{forecast pop}) &= a(\textit{forecast emp}) + b(\textit{base-yr pop})\\ (\textit{forecast emp}) &= c(\textit{forecast pop}) + d(\textit{base-yr pop})\end{aligned} \tag{3.25}$$

then a less straightforward procedure called two-stage least squares (2SLS) needs to be employed. The basic idea of 2SLS is to replace the endogenous explanatory variables $\mathbf{Y}$ in each equation with an estimated matrix $\hat{\mathbf{Y}}$ based on the regression of the variables in the $\mathbf{Y}$-vector on all of the predetermined (exogenous) variables, the $\mathbf{X}$ matrix in the model. This is referred to as stage 1 of the calibration. The second stage then involves ordinary least squares estimation of each Y_i based on $\hat{\mathbf{Y}}$ and $\mathbf{X}$.[6] For example, in the set of equations labeled 3.25 above, one can define the *forecast pop* and *forecast emp* as endogenous variables $\mathbf{Y}$ and *base-yr pop* as exogenous variable $\mathbf{X}$. Then stage one of 2SLS estimates a matrix of the two forecast variables $[\hat{\mathbf{Y}}_{pop}, \hat{\mathbf{Y}}_{emp}]$ based on the regression of these forecast variables on the base-year population. In the second stage, forecast variables, $\mathbf{Y}_{pop}$ and $\mathbf{Y}_{emp}$ are regressed against the estimated forecast-variables $[\hat{\mathbf{Y}}_{pop}, \hat{\mathbf{Y}}_{emp}]$ and the (*base-yr pop*) variable $\mathbf{X}$.

Mathematically, this process is shown by first moving all Y's to the left-hand side of the equation so that $\mathbf{D'y''} = \mathbf{B'x'} + \mathbf{A}$, where $\mathbf{D'}$ is the calibration-coefficient matrix, $\mathbf{B'}$ is the calibration-coefficient matrix and $\mathbf{A}$ is the disturbance or error vector. In this case $\mathbf{D'}$ is 2×2 matrix $\begin{bmatrix}1 & -a\\ -c & 1\end{bmatrix}$, $\mathbf{y}$ is 2×1 vector (*forecastpop, forecast emp*)^T, $\mathbf{B'}$ is the 2×2 matrix $\begin{bmatrix}b & 0\\ 0 & d\end{bmatrix}$, $\mathbf{x'}$ is a 2×1 vector (*base-yr pop, base-yr pop*)^T, and$\mathbf{A}$ is a 2×1 vector $(A_1\ A_2)^T$. Solving this equation results in $\mathbf{y''} = \mathbf{D'^{-1}B'x'} + \mathbf{D'^{-1}A}$, or what is commonly referred to as the reduced form of the original structure $\mathbf{y''} = \mathbf{Cx} + \mathbf{D'^{-1}A}$ where $\mathbf{C} = \mathbf{D'^{-1}B'}$. Then using ordinary least squares, estimates of the coefficients $\hat{\mathbf{C}}$, or $\mathbf{C}$, can be determined for each of the reduced form equations such that $\hat{\mathbf{Y}} = \hat{\mathbf{C}}^T\mathbf{X}$ where $\hat{\mathbf{Y}}$ is the estimated values of the endogenous variables. Here $\mathbf{C}$ is 2×2, $\mathbf{x}$ is 2×1, and $\hat{\mathbf{Y}}$ is 2×1. In the second stage, ordinary least squares is applied to the model

$$Y_i = \mathbf{a}^T\hat{\mathbf{Y}} + \mathbf{b}^T\mathbf{x}' + A_i \tag{3.26}$$

to find asymptotically unbiased estimates of the parameters $\mathbf{a}$ and $\mathbf{b}$. Here $\hat{\mathbf{Y}} = (Y_{pop}, \hat{Y}_{emp})^T$ and $\mathbf{a} = (a_1, a_2)$ and $\mathbf{b} = b$. In evaluating the significance of the model, similar measures to those employed in OLS can be used, *viz,* the Student-t test, the *F*-test, and the coefficient of multiple determination R^2. However, a word of caution is in order: Since both $\hat{\mathbf{Y}}$ and $\mathbf{y''}$ are used to compute R^2 in Equation 3.26, negative values can result. Therefore R^2 should not be used directly as a measure of the variation explained by the model. In addition, it follows from OLS that when there is high correlation between the explanatory variable, such as $\hat{\mathbf{Y}}$ and $\mathbf{x}'$ in Equation 3.26, the structural parameters **a** and **b** will be imprecise and have a high standard error.

C. Example of Two-Stage Least Squares

A linear urban model based on a set of simultaneous equations was developed for a twenty-three zone study area in central Berkshire, England. The calibration process involves running the regression model a number of times, testing for the significance of the variables, checking that the regression assumptions have not been violated, and interpreting the regression coefficients. The following equations set were postulated initially

$$\begin{aligned}\Delta N &= c_1 + a_1\Delta E^R + b_1\Delta E^B + b_2\Delta N + b_3 N + b_4 E^R + b_5 E^B + b_6 t' \\ \Delta E^R &= c_2 + a_2\Delta N + b_1\Delta E^B + b_8 N + b_9 E^R + b_{10} u\end{aligned} \tag{3.27}$$

Here N is zonal population, E^B is zonal basic employment, E^R is zonal service employment, t' is accessibility-to-employment, and u is accessibility-to-population. The Δ increments refer to changes over a five-year period. Using off-the-shelf econometric computer programs, the first stage of the calibration is to calculate the reduced form estimates for population and service employment changes from multiple regression equations expressing each as a function of all the exogenous variables. The second stage entails using these reduced form estimates as the explanatory variables on the right-hand sides of the simultaneous equations and performing multiple regression on each of the equations individually. This determines the coefficients c_1 to c_2, a_1 to a_2, and b_1 to b_{10}. At prediction, the coefficients of the reduced form equations and the coefficients of the simultaneous equations will be used in the same two-stage procedure to predict population and service employment changes over some future time period (Foot 1981).

Trial 1 of the calibration process produces the following result:

$$\begin{aligned}\Delta N &= 2514.51 + 27.702\Delta\hat{E}^R - 15.415\Delta E^B + 0.129N - 5.830E^R + 2.822E^B - 0.073t' \\ \Delta E^R &= 80.611 - 0.037\Delta\hat{N} + 0.555\Delta E^B - 0.0003N + 0.153E^R + 0.0045u\end{aligned} \tag{3.28}$$

where the reduced form estimates of population and service employment changes are denoted by ΔN and ΔE^R respectively. The R^2 for the first equation is 0.733, and the R^2 for the second is 0.998. The t-values for the coefficients associated with each explanatory variable in the first equation are respectively 0.973, 0.952, 1.344, 1.109, 1.529, and 0.503. Those for the second equation are 3.663, 12.614, 0.083, 38.25, and 2.647. To test the level of significance of the variables, the theoretical t-value at the 5 percent level for the first equation is $t = 2.114$, and for the second equation, $t = 2.106$.

It can be seen that many of the variables in both equations are not significant (particularly in the first equation), and this can largely be explained by the interrelationships between some of the variables which show up in the correlation matrix (Table 3.6). By inspection of this matrix and the level of significance of the coefficients, accessibility to employment was removed from the first

Table 3.6 CORRELATION MATRIX BETWEEN VARIABLES IN THE CENTRAL BERKSHIRE MODEL

	N	E^R	E^B	ΔN	ΔE^R	ΔE^B	u
E^R	0.6722						
E^B	0.7284	0.9028					
ΔN	0.1892	−0.3478	−0.0167				
ΔE^R	0.7042	0.9885	0.9113	−0.3026			
ΔE^B	0.5747	0.4634	0.6423	0.2719	0.5801		
u	0.8455	0.5413	0.5331	0.1559	0.5697	0.3929	
t'	0.8063	0.5342	0.4937	0.0859	0.5609	0.3613	0.9821

SOURCE: Foot (1981). Reprinted with permission.

equation and base-year population from the second equation, because of their high correlation with other variables in the equation. When the models were re-computed, service employment at the calibration year in the first equation was still not significant and, therefore, removed. The removal of these three variables lead to an increased level of significance of the other variables and, on re-computation, produced a simultaneous equation set containing only significant variables:

$$\begin{aligned} \Delta N &= 44.745 - 3.844\Delta\hat{E}^R + 2.372\Delta E^B + 0.144N + 0.924E^B \\ \Delta E^R &= -80.099 - 0.037\Delta\hat{N} + 0.555E^B + 0.152E^R + 0.0044u \end{aligned} \tag{3.29}$$

The first equation commands an R^2 of 0.700, while the second equation 0.998. The t-values in the first equation are 5.547, 2.174, 2.571, and 2.897; and for the second equation 4.625, 12.907, 50.667, and 3.508. It can be seen that service-employment change is almost perfectly reproduced by the model, and population change, significantly. The overall R^2 values have been reduced only slightly by removing the non-significant variables from the model. This latter, more parsimonious model satisfies the regression assumptions relating to the independence of the exogenous variables far better.[7]

The main problem with the final model is the negative coefficients, which in the first equation suggest that as service employment increases in a zone, population decreases. Similarly in the second equation, as population increases service employment decreases. This is due to the data used for calibration. With just the exogenous variables available in producing the reduced form estimates, the two-stage regression model cannot cope with extensive re-development that took place downtown in which shops and office buildings replaced substandard housing. Because of the large number of shops and offices downtown, the effects of this downtown re-development dominate over other zones in the entire study area. Exogenous variables relating to re-development must be included in the first stage of the regression to improve the explanation and provide more reasonable coefficients. In spite of this common problem among two-stage least squares, the calibrated model is statistically significant enough to be used for prediction in five-year increments, as long as the forecast is not over-extended into the future. For an example of this forecasting procedure, see the "Econometric Model" chapter in Chan (2005).

D. Maximum Likelihood

Another econometric technique to be discussed here is the maximum likelihood estimation procedure. It is a calibration procedure that estimates the unknown parameters by maximizing the probability, that the sample drawn is a true representation of the population, given the population distribution and sampling frame. An example of the gravity model is the best way to illustrate this model fitting technique. Suppose a consumer is choosing between two shopping malls ($k = 1$ or 2) to go to on Saturday morning. A sample of three shoppers ($n = 1, 2, 3$) has been included in a survey. Each shopper was asked about his individual travel time to a shopping mall k ($k = 1, 2$), τ_{nk}, and the final choice of the mall. The individual survey results are tabulated in Table 3.7. An examination of the table shows that the individuals surveyed chose a location mainly based on proximity.

Consider a model such as the following for an individual's discrete location decision as derived by consumers' surplus maximization in Chapter 2:

$$P(n, k) = \frac{\exp(\alpha_k \tau_{nk})}{\sum_{i=1}^{2} \exp(\alpha_i \tau_{ni})} \qquad k = 1, 2; n = 1, 2, 3 \tag{3.30}$$

A likelihood function L is defined as the probability that in a sample of three persons, one person chooses location 2 and two persons location 1. Thus the likelihood that the first two persons choose location 1 while the third location 2 is $P(1,1)P(2,1)P(3,2)$. There are three possible ways that the sample can have the one-person/location 2, two persons/location 1 split:

- the first person goes to shopping mall 2 while the second and third go to mall 1,
- the first and third go to mall 2 while the second goes to mall 1,
- the first and second go to mall 2 while the third goes to mall 1.

Table 3.7 DISAGGREGATE CALIBRATION OF A MAXIMUM LIKELIHOOD MODEL

Individual n	Time to location 1 τ_{n1} (Min)	Time to location 2 τ_{n2} (Min)	Locational choice k
1	5	7	1
2	4	6	1
3	6	4	2

SOURCE: Kanafani (1983). Reprinted with permission.

The likelihood function L now looks like

$$\frac{3!}{2!1!}P(1,1)P(2,1)P(3,2). \tag{3.31}$$

It is computationally convenient to take the logarithm of the likelihood function

$$\ln L = \ln 3 + \ln P(1,1) + \ln P(2,1) + \ln P(3,2). \tag{3.32}$$

which can be written as

$$K + \ln\frac{\exp(5\alpha_1)}{\exp(5\alpha_1) + \exp(7\alpha_2)} + \ln\frac{\exp(4\alpha_1)}{\exp(4\alpha_1) + \exp(6\alpha_2)} + \ln\frac{\exp(4\alpha_1)}{\exp(6\alpha_1) + \exp(4\alpha_2)} \tag{3.33}$$

where K is a constant. After collapsing of some terms

$$\begin{aligned} ln\ L = K + (4\alpha_2 + 9\alpha_1) - \{&\ln[\exp(5\alpha_1) + \exp(7\alpha_2)] + \\ &\ln[\exp(4\alpha_1) + \exp(6\alpha_2)] + \\ &\ln[\exp(6\alpha_1) + \exp(4\alpha_2)]\} \end{aligned} \tag{3.34}$$

The values of α's are simply determined by solving these two simultaneous equations.

$$\frac{\partial(\ln L)}{\partial\alpha_1} = 0; \quad \frac{\partial(\ln L)}{\partial\alpha_2} = 0 \tag{3.35}$$

These equations seek the values of α's that maximize the value function. The estimation typically involves the hill-climbing numerical technique, which is tangential to the development here and will be covered in Chapter 4, which summarizes prescriptive techniques.

Skipping over the computational details and getting at the results, the above two equations boil down to

$$9 - \frac{5\exp(5\alpha_1)}{\exp(5\alpha_1) + \exp(7\alpha_2)} + \frac{4\exp(4\alpha_1)}{\exp(4\alpha_1) + \exp(6\alpha_2)} + \frac{6\exp(6\alpha_1)}{\exp(6\alpha_1) + \exp(4\alpha_2)} = 0 \tag{3.36}$$

and

$$4 - \frac{7\exp(7\alpha_2)}{\exp(5\alpha_1) + \exp(7\alpha_2)} + \frac{6\exp(6\alpha_2)}{\exp(4\alpha_1) + \exp(6\alpha_2)} + \frac{4\exp(4\alpha_1)}{\exp(6\alpha_1) + \exp(4\alpha_2)} = 0 \tag{3.37}$$

Numerical solution of these two equations and two unknowns is performed, yielding $\alpha_1 = -14.434$ and $\alpha_2 = -14.211$. The reader should note the negative value of the α parameters. Compared with the OLS procedure discussed above, the maximum likelihood procedure typically is more efficient with data, but it requires knowledge of the underlying distribution of the basic random variable. It provides an unbiased estimator, rather than an asymptotically unbiased one. While the calibration procedure appears straightforward in this example, solutions to Equation 3.35 may be very difficult to find. When the errors are normally distributed, the maximum likelihood estimators of the regression coefficients are the least squares estimators.

X. AGGREGATE VERSUS DISAGGREGATE MODELING

The same shopping location example can be modeled using an aggregate format, which is simpler in many regards. Instead of addressing each individual's decision, the shoppers of the entire study area are modeled. Consider the case of three shopping centers from which the shoppers can choose. Table 3.8 shows the average travel cost and time to each of these centers, as well as the number of patrons that end up there. The objective is to calibrate an aggregate, instead of disaggregate, location choice model. The model specification will look like

$$P_k = \frac{\exp(a\tau_k + bc_k)}{\Sigma_i \exp(a\tau_i + bc_i)} \tag{3.38}$$

where τ_k and c_k are the travel time and cost via mode k, and a, b are calibration constants. Notice that instead of location-specific calibration parameters α_1 and α_2, a single set is used across all centers, indicating a homogeneous behavior among the shoppers. The above is often referred to as a multinomial logit model. The maximum likelihood function looks like

$$L = \frac{100!}{50!\ 40!\ 10!} P_1^{50} P_2^{40} P_3^{10}.$$

Let X be the denominator for P_1, P_2, and P_3 (see Equation 3.38). Then ln L = $50(15a + 3b) + 40(10a + 4b) + 10(20a + 7b) - 100 \ln X$ in which values for τ_k and c_k are obtained from Table 3.8.

Table 3.8 DATABASE FOR CALIBRATING AN AGGREGATE LOCATION CHOICE MODEL

Shopping center k	Average time τ_k	Average cost c_k	No of patrons at center k
1	15	3	50
2	10	4	40
3	20	7	10

$$\frac{\partial(lnL)}{\partial a} = 1350 - (100/X)\ [15\exp(15a + 3b) + 10\exp(10a + 4b) + 20\exp(20a + 7b)] = 0 \quad (3.39)$$

and

$$\frac{\partial(lnL)}{\partial a} = 380 - (100/X)[3\exp(15a + 3b + 4\exp(10a + 4b) + 7\exp(20a + 7b)] = 0 \quad (3.40)$$

Solution of these simultaneous equations yields $a = -0.02868$ and $b = -0.36640$. Again, the readers should note the negative signs for the parameters a and b.

There are several implications from aggregate, rather than disaggregate, modeling. The straightforward one is that the calibration procedure is more simple. The more noteworthy one is that aggregate and disaggregate modeling have very different behavioral assumptions. This point is best shown by the following replication test, where the calibrated model is used to reproduce the observed data, as a descriptive model should. To show the replication test, one should be aware of the fact that a logit choice model discussed above can be represented as

$$P_1 = \frac{1}{1 + \exp\Delta} \quad \text{and} \quad P_2 = \frac{\exp\Delta}{1 + \exp\Delta} \quad (3.41)$$

for the two shopping center cases, where $\Delta = \beta\ \delta c + \gamma\,\delta\tau + \alpha$ in which $\delta\tau = \tau_2 - \tau_1$ and $\delta c = c_2 - c_1$. Suppose a disaggregate model is calibrated with $\beta = -293.2$ $\gamma = -71.3$ and $\alpha = 1.93$ based on time-unit of hours and cost in $\$ \times 10^{-2}$. This means that for the data shown in Table 3.9, an entry of 17 minutes should be translated to 0.293 hours and \$2.16 should be translated to 0.0216 before they are substituted into the model formulas. Using the model consistently in disaggregate prediction will yield $\Delta_1 = -293.2(.02) - 71.3(.0833) + 1.93 = -9.87$, $\Delta_2 = -4.01$ and $\Delta_3 = 2.09$. These values, when substituted into Equation 3.41, yield P_2 of 0, 0, and 1.000 for shoppers 1, 2, and 3 respectively (or locational decisions of 2, 2, 1). The average of these three P_2s is 0.333, which agrees with the observed data in Table 3.9:

Table 3.9 REPLICATION TEST DATA FOR LOGIT MODEL

Shopper	Time to location 1 τ_1 (Min)	Time to location 2 τ_2 (Min)	$\delta\tau = \tau_2 - \tau_1$ (Min)	Cost to location 1 c_1 (\$)	Cost to location 2 c_2 (\$)	$\delta c = c_2 - c_1$ (\$)	Locational decision
1	25	30	−5	3.00	1.00	2	1
2	10	15	−5	1.00	1.00	0	1
3	50	40	10	5.00	1.00	4	2
Average	—	—	0	—	—	2	—

$$P_2 = \frac{(2\ shopper\ at\ 2)}{(a\ total\ of\ 3\ shoppers)} = 0.333$$

On the other hand, misuse of the model by using average travel time and cost will lead toward totally erroneous predictions. Thus substituting aggregate data $\Delta = -293.2(0.02) - 71.3(0) + 1.93 = -3.93$ into Equation 3.41 will result in $P_1 = 0.02$ and $P_2 = 0.98$, which is far from reality. Thus, care must be exercised in the calibration and consistent use of aggregate versus disaggregate models. As long as aggregation across individuals is handled with care, it need not be a major source of error in the forecasting process.

XI. THE GRAVITY MODEL REVISITED

We have described above the calibration of a location choice model as represented in Equations 3.30 and 3.38. It will be shown here that the aggregate version of the two models can be developed from first principles other than consumers' surplus maximization (as discussed in Chapter 2), and they can be calibrated with a method other than maximum likelihood. We start with the functional form $V_{ij} = V(\mathbf{S}_{ij}, \mathbf{A}_j)$ where $\mathbf{S}_{ij}$ is the vector of level-of-service variables between i and j as measured in accessibility. (Recall that accessibility is an inverse function of travel cost, travel time, and other spatial-separation metrics.) $\mathbf{A}_j$ is a vector of socioeconomic variables representing such activities as population and employment.

A. Singly Constrained Gravity Model

Let us use F_{ij} to denote an accessibility factor, defined as an inverse function of travel cost in a form such as $\exp(-bC_{ij})$ and C_{ij}^{-b}. Let us also use V_j to denote the attraction at destination j, where the attraction may be employment opportunities, or in this case simply the trips terminating at the destination zone V. A model can now be constructed bearing the form $V_{ij} = MV_iF_{ij}V_j$ where M is a calibration constant. Since the sum of the originating trips have to add up to the production, or $\Sigma_j V_{ij} = V_i$, we can write $\Sigma_j MV_iF_{ij}V_j = V_i$. Canceling the V_i term from both sides of the equation and extracting the calibration constant M from the summation sign, we have $M\Sigma_j F_{ij}V_j = 1$ or $M = 1/(\Sigma_j F_{ij}V_j)$. Substituting this calibration constant M back to the original equation, we have

$$V_{ij} = \frac{V_i F_{ij} V_j}{\Sigma_j F_{ij} V_j}$$

which is the familiar singly constrained gravity model.

Consider a region consisting of four zones 1, 2, 3, and 4. Residents in zones 1 and 2 are considering shopping at zones 2, 3 and 4. The existing travel pattern is represented in Table 3.10. As can be seen, there are 1000 potential trip productions emanating from zone 1 and 1400 from zone 2. The trip

Table 3.10 EXISTING INTERZONAL TRAVEL

From/to	Zone 1	Zone 2	Zone 3	Zone 4	V_i
Zone 1		500	200	300	1000
Zone 2		800	100	500	1400
Zone 3					
Zone 4					
V_j		1300	300	800	2400

SOURCE: Dickey (1983). Reprinted with permission.

attractions at zones 2, 3, and 4 are 1300, 300, and 800 respectively. The travel costs (as represented in minutes of travel times) between the zones, C_{ij}, are shown in Table 3.11. A travel accessibility function of $F_{ij} = C_{ij}^{-2}$ is assumed, or the calibration parameter b is set to 2 initially. This means a set of $F(C_{ij})$s that appears as follows:

C_{ij}	3	5	8	10
$F(C_{ij})$	0.111	0.0400	0.0156	0.0100

Now the interzonal trips can be estimated by

$$V_{12} = 1000\left[\frac{(1300)(0.0156)}{(1300)(0.0156) + (300)(0.0400) + (800)(0.0100)}\right] = 503 \quad (3.42)$$

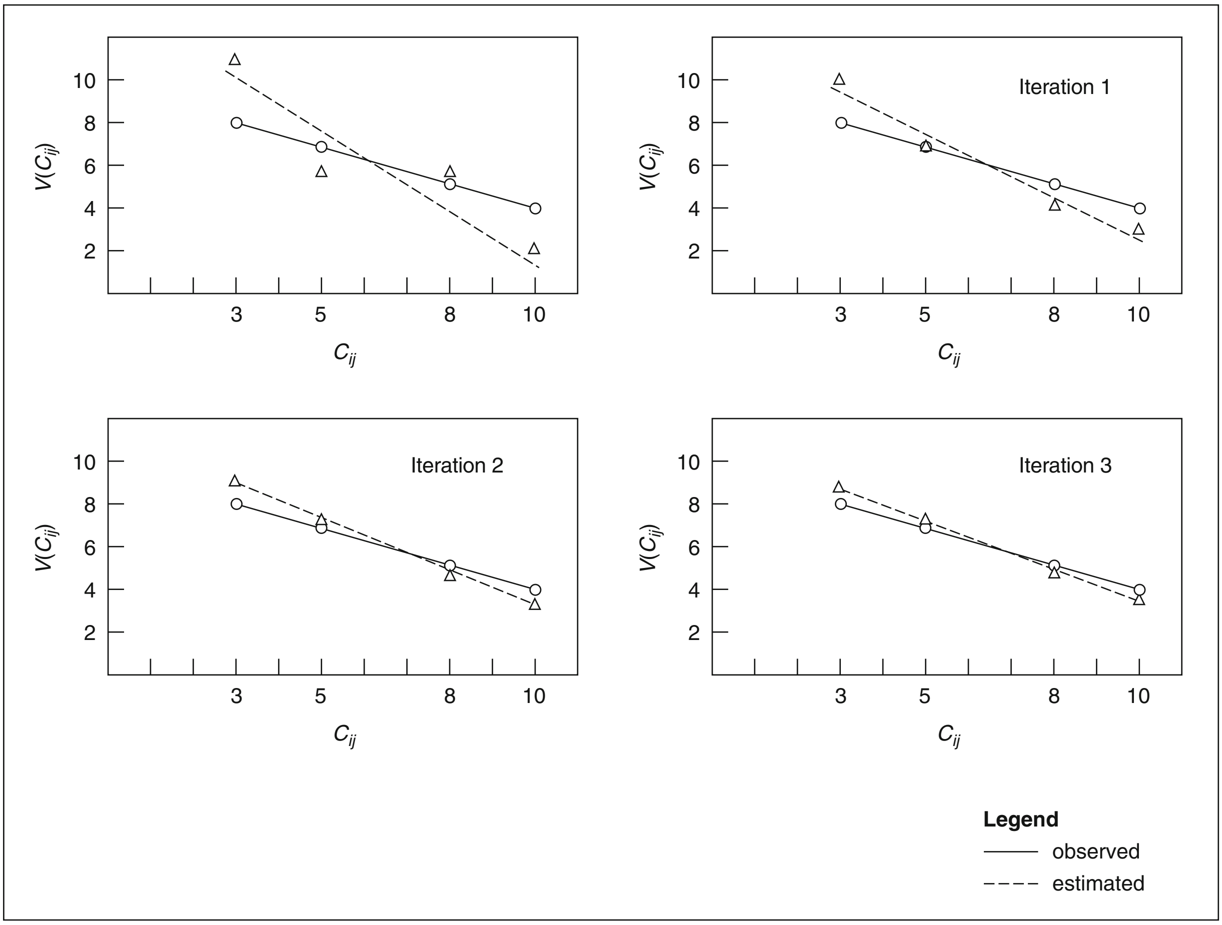
Similarly, $V_{13} = 298$, $V_{14} = 199$, $V_{22} = 1127$, $V_{23} = 23$, and $V_{24} = 250$. Since $V_{12} + V_{22} = 1630 \neq 1300$, calibration of the model is necessary in order to replicate the existing data more closely. The need for calibration is best shown by a trip distribution plot such as Figure 3.17, where trips of a certain duration, say 3, 5, 8, and 10 minutes are plotted. It can be seen that the observed curve is significantly different from the estimated. To bring the estimated and observed trip distribution curves together, the accessibility factors $F(C_{ij})$ can be adjusted by scaling the

Table 3.11 INTERZONAL TRAVEL TIMES (IN MINUTES)

From/to	Zone 1	Zone 2	Zone 3	Zone 4
Zone 1	3	8	5	10
Zone 2	8	3	10	5
Zone 3	5	10	3	20
Zone 4	10	5	20	3

SOURCE: Dickey (1983). Reprinted with permission.

Figure 3.17 TRIP DISTRIBUTION PLOTS



points on the curve according to the observed data. For example,

$$F'(3) = F'_{22} = F_{22}(800/1127) = (0.1111)(800/1127) = 0.0789$$
$$F'(5) = F'_{13} = F'_{24} = (0.04)(700/548) = 0.0511$$
etc.

This also yields $F'(8) = F'_{12} = 0.0155$ and $F'(10\ ) = F'_{14} = F'_{23} = 0.0180$. From these accessibility factors, new estimates can be made on interzonal travel. For example,

$$V'_{12} = 1000\left[\frac{(1300)(0.0155)}{(1300)(0.0155) + (300)(0.0511) + (800)(0.0180)}\right] = 404 \qquad (3.43)$$

Similarly, it can be shown that $V'_{13} = 307$, $V'_{14} = 289$, $V'_{22} = 965$, $V_{23} = 51$, and $V'_{24} = 385$. Based on these estimated trips, the trip distribution plot is shown again in Figure 3.17.

Table 3.12 CHI-SQUARE TEST

Time Interval (Min)	3	5	8	10
Observed y_i	800	700	500	400
Estimated $\hat{y}_i$	857.28	707.90	457.40	377.00
$(y_i - \hat{y}_i)$	−57.28	−7.90	42.60	23.00
$(y_i - \hat{y}_i)^2$	3281.00	62.41	1814.76	529
$(y_i - \hat{y}_i)^2/\hat{y}_i$	3.83	0.09	3.97	1.40
$\Sigma_i(y_i - \hat{y}_i)^2/\hat{y}_i$	3.83	3.92	7.88	9.29

The process is repeated until the third iteration, when the two distribution curves seem to agree with one another, as shown in Figure 3.17. At this iteration, $V'''_{12} = 457$, $V'''_{13} = 245$, $V'''_{14} = 298$, $V'''_{22} = 857$, $V''_{23} = 79$, and $V'''_{24} = 463$. The goodness of fit between the two curves can be shown formally by the chi-square test. To explain this test, the step-by-step computation is organized around Table 3.12. The degree of freedom is $n - 1 = 4 - 1 = 3$. From a chi-square table of any statistics text, χ^2 to a 0.95 significance level with 3 degrees of freedom is $\chi^2(0.05{,}3) = 7.815$. The fact that chi-square statistic of 9.29 from Table 3.12 is bigger than 7.815 means that the fit between the trip distribution curves is not statistically significant.

Recall that we hypothesized a coefficient of $b = 2$ in the accessibility factor $F(C_{ij}) = C_{ij}^{-b}$ initially for the model, but found that it was not giving the best fit to the data in the trip distribution curve initially. Over the three iterations, we have modified sufficiently the calibration parameter b by adjusting the F values. At the termination of the algorithm, we have a set of F values, from which the final calibrated parameter b can be recovered. To do this, we first take the logarithm of $F(C_{ij}) = C_{ij}^{-b}$: $\ln F = -b \ln C$. $\ln F$ is then regressed against $\ln C$ using the following set of data, with the stipulation that the regression line will go through the origin.[8] The slope of the regression line is then simply the b value we are looking for.

	$\ln F$	$\ln C$
$F(3) = 0.0583$	−2.8422	1.0986
$F(5) = 0.0512$	−2.9720	1.6094
$F(8) = 0.0221$	−3.8122	2.0794
$F(10) = 0.0234$	−3.7550	2.3026.

Result of the regression shows that $b = 1.874$ at an R^2 of 0.9958. In other words, the slope of the trip distribution curve should be gentler than first hypothesized, as illustrated in Figure 3.17.

B. Doubly Constrained Model

It is quite obvious that the above model is not easy to calibrate since a number of ad hoc procedures need to be strapped together to achieve the desired goodness of fit. A simpler alternative is to formulate a doubly constrained model that explicitly takes into account the constraints placed on the number of trip attractions in addition to the number of productions (Oppenheim 1980). Consider the following model for a study area consisting of n' zones $V_{ij} = k_i l_j V_i V_j F(C_{ij})$ such that

$$\begin{aligned} \sum_{i=1}^{n'} V_{ij} &= V_j \qquad j = 1, \ldots, n' \\ \sum_{j=1}^{n'} V_{ij} &= V_i \qquad i = 1, \ldots, n' \end{aligned} \tag{3.44}$$

Notice that instead of one calibration constant M, two constants k_i and l_j are introduced. By substitution of these constraints into the initial equation for the model, we have

$$\begin{aligned} \sum_{i=1}^{n'} k_i l_j V_i V_j F(C_{ij}) &= V_j \qquad j = 1, \ldots, n' \\ \sum_{j=1}^{n'} k_i l_j V_i V_j F(C_{ij}) &= V_i \qquad i = 1, \ldots, n' \end{aligned} \tag{3.45}$$

These reduce to

$$\begin{aligned} l_j \sum_{i=1}^{n'} k_i V_i F(C_{ij}) &= 1 \qquad j = 1, \ldots, n' \\ k_i \sum_{j=1}^{n'} l_j V_j F(C_{ij}) &= 1 \qquad i = 1, \ldots, n' \end{aligned} \tag{3.46}$$

after canceling V_j on both sides of the first equation, and likewise for V_i of the second. The calibration constants can now be determined:

$$\begin{aligned} l_j &= \frac{1}{\sum_{i=1}^{n'} k_i V_i F(C_{ij})} = 1 \qquad j = 1, \ldots, n' \\ k_i &= \frac{1}{\sum_{j=1}^{n'} l_j V_j F(C_{ij})} = 1 \qquad i = 1, \ldots, n' \end{aligned} \tag{3.47}$$

Notice the two equation sets are coupled together, in that k appears on the right-hand side of the first equation set, and l appears on the right-hand side of the second. An iterative solution strategy is anticipated. A numerical example will make this clear.

Example
Given these interzonal travel times

$$[C_{ij}] = \begin{bmatrix} 2 & 4 & 8 \\ 5 & 1 & 7 \\ 7 & 6 & 3 \end{bmatrix}$$

Table 3.13 OBSERVED INTERZONAL TRAVEL OF A DOUBLY CONSTRAINED MODEL

From/to	$j = 1$	$j = 2$	$j = 3$	V_i
$i = 1$	1,800	3,100	100	5,000
$i = 2$	3,100	1,500	400	5,000
$i = 3$	15,100	25,400	4,500	45,000
V_j	20,000	30,000	5,000	55,000

SOURCE: Oppenheim (1980). Reprinted with permission.

the following accessibility factors can be derived for a particular functional form and an assumed value of the calibration constant b:

$$[F(C_{ij})] = \begin{bmatrix} 1.472 & 2.165 & 1.172 \\ 2.052 & 0.607 & 1.480 \\ 1.480 & 1.792 & 2.008 \end{bmatrix}$$

The observed values of interzonal travels are shown in Table 3.13.

We wish to solve the six equations and six unknowns for k_1, k_2, k_3, l_1, l_2, and l_3 as represented by Equation 3.47 where $n' = 3$ in this case. Suppose we start with the arbitrary values of 1 for the k's. Substituting 1's in the formulas will yield $l_1 = 0.1187 \times 10^{-4}$, $l_2 = 0.1058 \times 10^{-4}$, $l_3 = 0.0965 \times 10^{-4}$. Now substitute these l values into the formulas for the k's in Equation 3.47, one will find that these new values for the k's are no longer l's. We continue this process until a consistent set of ks and ls are obtained, as shown in Table 3.14. It can be seen that we obtain convergence within four iterations.

Table 3.14 CALIBRATION OF A DOUBLY CONSTRAINED MODEL

	Iteration Number			
	1	**2**	**3**	**4**
k_1	1	0.9148	0.9125	0.9125
k_2	1	1.3312	1.3437	1.3437
k_3	1	0.9833	0.9824	0.9824
l_1	0.1187[a]	0.1164	0.1164	
l_2	0.1058	0.1073	0.1073	
l_3	0.0965	0.0961	0.0961	

[a] All these nine l values are to be multiplied by 10^{-4}. For example, 0.1187 is actually 0.1187×10^{-4}.

SOURCE: Oppenheim (1980). Reprinted with permission.

Table 3.15 ESTIMATED INTERZONAL TRAVELS IN A DOUBLY CONSTRAINED MODEL

From/to	$j = 1$	$j = 2$	$j = 3$	V_i
$i = 1$	1,563 (−15.2)[a]	3,178 (+2.5)	257 (+61.1)	4,998
$i = 2$	3,209 (+3.4)	1,313 (−14.3)	478 (+16.3)	5,000
$i = 3$	15,238 (+1.0)	25,498 (+0.4)	4,265 (−5.51)	45,001
V_j	20,010	29,989	5,000	54,999

[a] Numbers in parentheses indicate the percentage errors between observed and estimated interzonal travel.

SOURCE: Oppenheim (1980). Reprinted with permission.

Based on these values of k and l, $\hat{V}_{ij}$s can be estimated as shown in Table 3.15. Also shown in the same table is the percentage error between estimated ($\hat{V}_{ij}$) and observed interzonal travel ($V_{ij}{}^*$): $(V_{ij} - \hat{V}_{ij})/V_{ij}$. To reduce the error further, another functional form for the accessibility factor may be in order, either by changing from a power function C_{ij}^{-b} to exponential function $\exp(-bC_{ij})$ or vice versa (among other possible functional forms), or changing the initial value of b. Such a decision can be assisted by examining the plots of the trip distribution curves, as illustrated in the singly constrained gravity model example. ■

XII. SPATIAL INTERACTION

As can be seen from the gravity model calibration, one of the major steps in spatial-temporal analysis is to enrich our information about the study area based on partially observable data. We have seen from the numerical examples above that an $n' \times n'$ matrix of trip movements is to be constructed from given row and column sums, often referred to as the trip productions and attractions (or more properly the origin trips and destination trips). In this case, we wish to estimate n'^2 pieces of information from $2n'$ pieces of data when certain statements can be made about travel behavior, as manifested in the trip distribution function showing the relative trip lengths in the area. In short, we wish to provide more complete activity distribution information from scanty observations. There are two formal methods to do this: minimum information theory and entropy maximization.

A. Information Theory

Here, let us concentrate on a facility location example. Suppose a firm is about to locate in one of the n' zones of a region. A land developer has studied the firm and its needs, and concludes that the probability of the firm locating in zone 1 is Q_1, in zone 2 is Q_2, . . . , and more generally, of locating in zone i is Q_i. The number Q_i is the developer's guess about the likelihood of the firm locating in zone i. Alternatively expressed, the ratio $Q_i/(1 - Q_i)$ is the odds on the firm choosing zone i. Clearly, probabilities cannot be negative (so the non-negativity requirement applies: $Q_i \geq 0$) and the firm must locate somewhere (so the sum of the subareal shares must be unity: $\Sigma_i Q_i = 1$). Perhaps the developer then receives inside information that one member of the board favors a particular zone, say j. The developer is therefore forced to revise his/her estimates so that the estimated probability of the firm locating in zone i is now P_i. The insider message has evidently caused the developer to change his or her mind about the likely outcome of the event and has therefore imparted improved information (Webber 1984; Gonzales and Woods 1992). The question is: How much more information than before?

The developer starts with the probability distribution $\mathbf{Q} = (Q_1, Q_2, \ldots, Q_{n'})$, and changes this opinion to $\mathbf{P} = (P_1, P_2, \ldots, P_n')$. Let the extra information contained in the insider message which updates the probabilities $\mathbf{Q}$ to $\mathbf{P}$ be denoted by $I(\mathbf{P}; \mathbf{Q})$. Five desiderata are now associated with the measure $I(\mathbf{P}; \mathbf{Q})$ and we need to address these five specifications for $I(\mathbf{P}; \mathbf{Q})$. First, the two probability distributions may in fact be one and the same ($\mathbf{P} = \mathbf{Q}$). In this case, the message does not change the developer's mind. This means the information is worthless or $I(\mathbf{P}; \mathbf{Q}) = 0$.

Second, it is reasonable to require that the information conveyed by the message does not depend on the order in which zones are listed or labeled, say from zone 1 to zone n'. In other words, it does not matter whether, for instance, the downtown zone is labeled as zone 1, zone 2, or zone 3 and so on.

Third, it is required that the metric $I(\mathbf{P}; \mathbf{Q})$ be continuous. Thus, if the message has only a small effect in updating the probabilities (or $\mathbf{P}$ is very similar to $\mathbf{Q}$), only a little information is gained in the process of updating $\mathbf{P}$ to $\mathbf{Q}$. In other words, $I(\mathbf{P}; \mathbf{Q})$ is very small or nearly zero, slight differences in probability distributions are associated with marginal information.

Fourth, suppose that the developer has only the knowledge on the zones in which the firm will not locate. Lacking further information, the developer believes that each of the remaining zones is equally likely to be chosen. Three special cases can be defined for this situation:

(a) One initially believed the n' zones were feasible, but reduced this to $(K + 1)$ when given a message, where $0 < K < n'$. This means that $\mathbf{Q} = (1/n' + 1/n', \ldots, 1/n')$, and $\mathbf{P} = (1/(K + 1), 1/(K + 1), \ldots, 1/(K + 1))$ where $P_i \geq Q_i$.

(b) One believed that n' zones were feasible, but reduced them to $\boldsymbol{K}$ upon receipt of insider information: $\mathbf{Q} = (1/n', 1/n', \ldots, 1/n')$, $\mathbf{P} = (1/K, 1/K, \ldots, 1/K)$. Here $P_i > Q_i$.

(c) One initially believed that $(n' + 1)$ zones were feasible, but reduced that number to K when given insider information (remembering $K < n'$): $\mathbf{Q} = (1/(n' + 1), 1/(n' + 1), \ldots, 1/(n' + 1))$, $\mathbf{P} = (1/K, 1/K, \ldots, 1/K)$. Here $P_i >> Q_i$.

In terms of the number of zones taken out of contention by the insider's message, (a) has received the least information and (c) the most, and I($\mathbf{P}$; $\mathbf{Q}$) is required to satisfy this ordering of information contents. In other words, as one progresses from (a) to (c), P_i becomes larger (or the location of the firm becomes more definite) as more information is available.

Fifth and last, the zones may be classified into two groups: 1, 2, . . . , K, and $K + 1$, $K + 2$, . . . , n'. The location question then becomes: what are the probabilities that the firm will locate in group 1 or group 2, and given that any one group is chosen, what is the probability that the firm will choose a particular zone in that group? Let (Q_1*, Q_2*) and (P_1*, P_2*) be the prior and posterior probabilities of choosing each group, and let $\mathbf{Q}_1$ and $\mathbf{Q}_2$ (or $\mathbf{P}_1$ and $\mathbf{P}_2$) be the prior (or posterior) probabilities of choosing a zone within each group. Then an insider's message provides information about group membership (or changes **Q*** to **P***) and about specific zone location given the group has been identified (or changes $\mathbf{Q}_1$ and $\mathbf{Q}_2$ to $\mathbf{P}_1$ and $\mathbf{P}_2$ respectively). In this case, the expected total amount of information is $I(P_1^*, P_2^*; Q_1^*, Q_2^*) + P_1^* I(\mathbf{P}_1; \mathbf{Q}_1) + P_2^* I(\mathbf{P}_2, \mathbf{Q}_2)$, or the combined information of group identification and zone location.

These five desiderata are posed on the measure of information. Together the five uniquely specify the mathematical measure of the information provided by the insider message that changes probabilities from **Q** to **P**:

$$I(\mathbf{P}; \mathbf{Q}) = \sum_{i=1}^{n'} P_i \ln \frac{P_i}{Q_i} \tag{3.48}$$

The fundamental premise of information theory is that the generation of information can be modeled as a probabilistic process that can be measured in a manner that agrees with intuition. In accordance with this supposition, a random event A that occurs with probability $P(A)$ is said to contain $I(A) = \ln(1/P(A)) = -\ln P(A)$ units of information. The quantity $I(A)$ is often called the **self-information** of A. Generally speaking, the amount of information attributed to event A is inversely related to the probability of A. If $P(A) = 1$ (that is, the event occurs with certainty), $I(A) = 0$ and no information is attributed to it. In other words, because no uncertainty is associated with the event, no information would be imparted by communicating that the event has occurred. However, if $P(A) = 0.99$, communicating that A has occurred conveys some small amount of information. Communicating that A has not occurred onveys much more information, because this outcome is much less likely, $P(\sim A) = 0.01$. Thus in Equation 3.48, $\ln(P_i/Q_i) = -(\ln P_i - \ln Q_i)$ is the information gained from the insider message about locating the firm in zone i. Weighing each zone by the current probability P_i and summing the zonal information gain over n' zones provides the mathematical expression for minimum discrimination information over the entire study area. Most importantly, it can be shown that this expression possesses all the five desired properties outlined above.

A common way to operationalize the metric in Equation 3.48 is the entropy measure[9]. If **Q** is a uniform distribution in Equation 3.48 (i.e., if $Q_i = 1/n'$ for every $i = 1, 2, \ldots, n'$) then

$$I\,(\mathbf{P}, \mathbf{Q}) = \sum_i P_i \ln \frac{P_i}{1/n'}, = \sum_i P_i \ln(P_i n') = \sum_i P_i \ln P_i + \sum_i P_i \ln n' = \sum_i P_i \ln P_i + \ln n' \tag{3.49}$$

Consider a firm choosing a location among n' zones to open business. A priori, it is believed that the probability that a firm should be located in zone i is Q_i, for each $i = 1, 2, \ldots, n'$. Some structural or aggregate data are now obtained that describe the locational decision at hand; call these data D'. The problem is to describe the spatial distribution of firms (or the probability that any one firm locates at each zone), given that D' alone are insufficient to provide such detailed information. The logical inferential method of solving this problem is called **minimum information principle**. As beliefs are changed from **Q** to **P**, it reflects that an amount of extra information is gained to effect the change; different **P**'s correspond to different amounts of information. The minimum information principle requires that a value of **P** is chosen that minimizes the apparent information given by the data D', but subject to the requirement that **P** is consistent with D'. Thus the method asserts that the phenomena should be described in the way which deviates least from the original beliefs, apart from the modifications dictated by D'.

B. Entropy

Now let the journey-to-work trip distribution be $\mathbf{P} = \mathbf{V} = [V_{ij}]$ be chosen with minimum information from a priori distribution $\mathbf{Q} = [Q_{ij}]$ (Putman 1978; Cesario 1975). Notice here, that without violating any of the arguments above, V_{ij} and Q_{ij} are no longer probabilities. In terms of information theory, this can be represented as choosing **V** to minimize

$$I(\mathbf{V};\mathbf{Q}) = \sum_{i=1}^{n'} \sum_{j=1}^{|J|} V_{ij} \ln \frac{V_{ij}}{Q_{ij}} \tag{3.50}$$

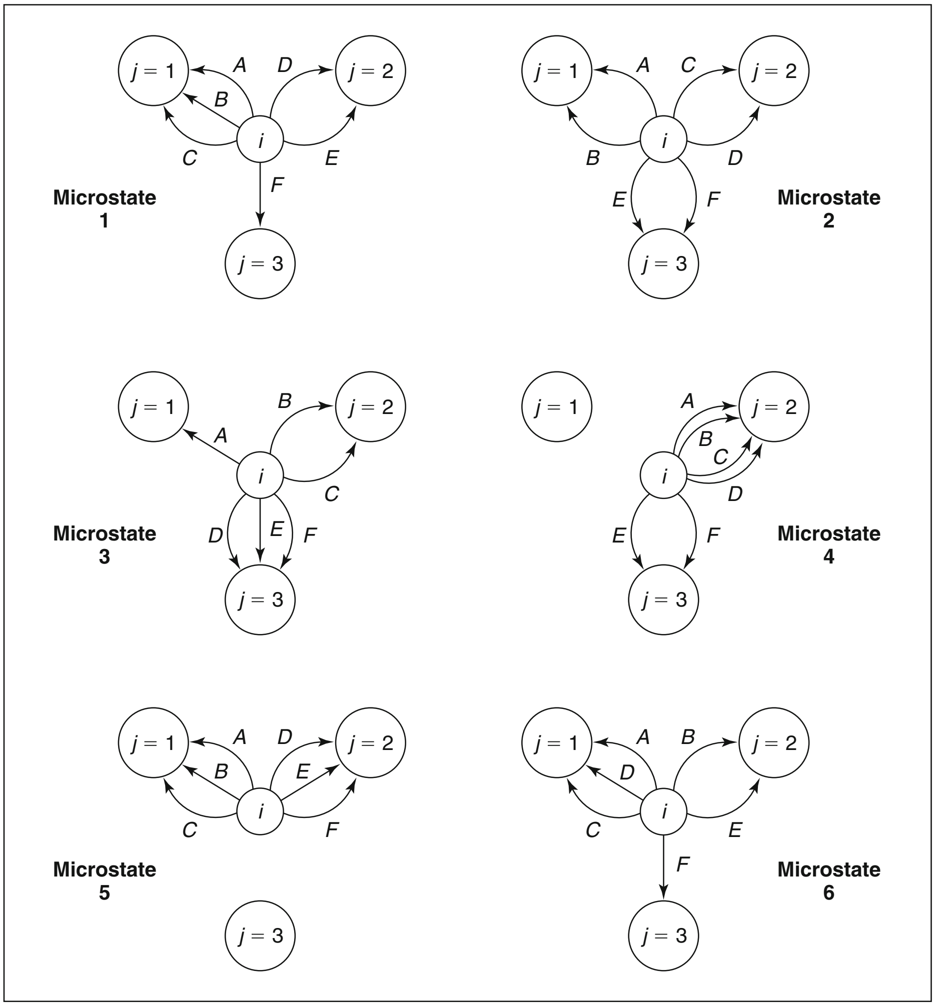
subject to given data D'. Here n' stands for the number of origin zones and $|J|$ the number of destination zones. To the extent that $\mathbf{Q} = [Q_{ij}]$ is given, the above expression is equivalent to

$$\text{Min } \Sigma_i \Sigma_j V_{ij} \ln V_{ij} - \ln Q_{ij} \qquad \text{or} \qquad \text{Min } \Sigma_i \Sigma_j V_{ij} \ln V_{ij} \tag{3.51}$$

To interpret this, let us examine a simple example due to Senior (1973). Imagine six employed persons living in one residential zone $i = 1$, and commutes to three work zones $j = 1, 2, 3$. Suppose that the six workers are named *A, B, C, D, E*, and *F*. We may now specify the origins and destinations of the work trips for each worker. Each possible, fully described, system of (a) one origin, (b) three destinations, and (c) six total work trips with their specified origins and destinations may be called a **microstate** of the system. Six of these possible microstates are shown in Figure 3.18. There are obviously many more since there are very many such microstates of even this simple system.

Let us now consider microstate 1 where the number of trips between i and $j = 1$ is 3; the number between i and $j = 2$ is 2; and the number between i and $j = 3$ is 1. Microstate 6 may also be seen to have this same distribution of trips: from i to $j = 1$ there are 3 trips, from i to $j = 2$ there are 2, and from i to $j = 3$, there is 1. Clearly there are many microstates that could be drawn that would have this same arrangement of total trips. This particular arrangement of zone to zone trips, if described independently of which worker is making which trip, may be called the **mesostate** of the system. Four mesostates of the system are shown in Figure 3.19. Comparing Figures 3.18 and 3.19, it can be seen that the microstates 1 and 6 are two possible manifestations of mesostate *A*. Microstate 2 is a possible manifestation of mesostate *B*, but microstate 5 is *not* a

Figure 3.18 SYSTEM MICROSTATES

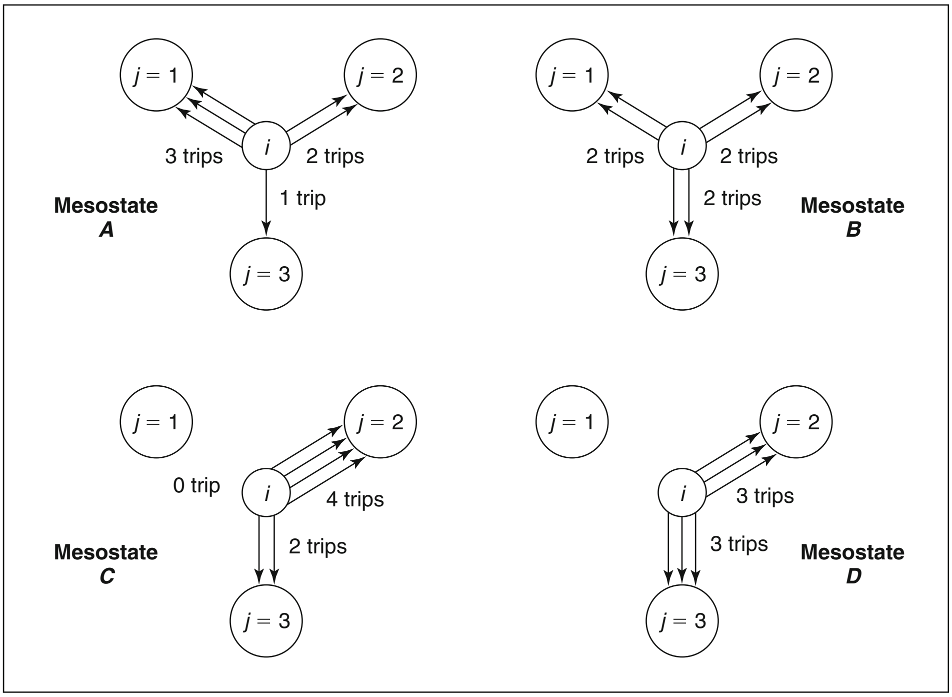


SOURCE: Putman (1978). Reprinted with permission.

possible manifestation of mesostate *D*. Thus each mesostate describes a specific set of possible microstates.

If we now consider that there might be several residential zones in addition to the one which has been used in this example, then a more aggregate description of the system would be the total trips leaving each origin and the total trips arriving at each destination. Let us assume that two workers live in zone $i = 2$, and four workers live in zone $i = 3$ in addition to the six already defined as living in $i = 1$ (and we equate one worker with one work trip as we have been doing). Further assume that these additional workers are named *G, H, I, J, K'*, and *M*. A microstate of this newly expanded system would be a list of the origins and destinations of the work trips for each of the 12 workers, $V_{ij}(k)$; $k = A, B, \ldots, M$. A mesostate of this system would be a list of the total number of work trips from each origin zone to each destination zone Q_{ij} or V_{ij}. Finally, a **macrostate** of this expand system is a list of the total trips leaving each origin and the total trips

Figure 3.19 SYSTEM MESOSTATES



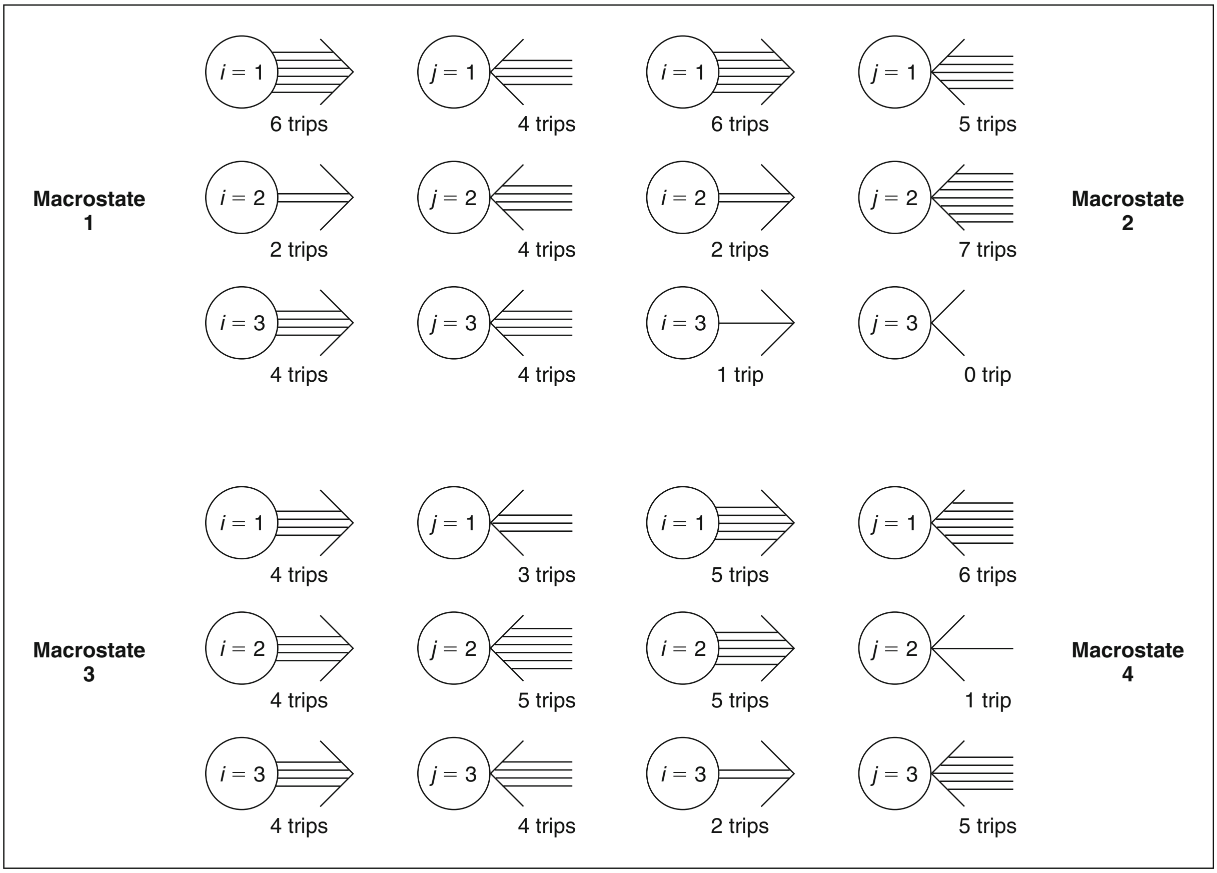
SOURCE: Putman (1978). Reprinted with permission.

arriving at each destination, V_i. Figure 3.20 shows four macrostates of the expanded system.

Referring to Figure 3.20, macrostates 1 and 2 with 6 trips leaving $i = 1$ contain all the previous examples of microstates and mesostates. Macrostates 2 and 3, with the trips leaving $i = 1$ not equal to six, correspond to other system states which do not include the microstates and mesostates given as examples. We should also note in passing that one could have defined a macrostate for the example of a single origin used at the start of this discussion. This would have been, in a sense, a degenerate case, as the trips leaving the single origin would lways have been equal to six. The microstates as defined here in this discussion correspond to a disaggregate model as referred to in Section V; mesostates correspond to aggregate modeling; and macrostates are the given conditions for calibrating a gravity model. The entropy formulation deals with the meso- and macrostates and requires two key assumptions. First, all microstates are assumed to be equally probable. Second, the most likely mesostate or macrostate is assumed to be the one with the greatest number of possible microstates. We refer to this second assumption as **entropy maximization.**

We may now develop a spatial interaction model for the mesostate level using entropy, rather than gravity model formulation. The given information could consist of the origin trips V_i and the destination trips V_j where $\Sigma_j V_{ij} = V_i$ and $\Sigma_i V_{ij} = V_j$, or the total number of trips $Q = \Sigma_i V_i = \Sigma_j V_j = \Sigma_{i,j} V_{ij}$. Let us examine the microstates $V_{ij}(k)$ for a given Q. For the example in Figure 3.17, $i = 1$, $j = 1$, 2, 3 and $V = 6$, the microstates consist of $V_{11}(k)$, $V_{12}(k)$ and $V_{13}(k)$. Stripping the traveler designation k, $V_{11} + V_{12} + V_{13} = Q$. V_{11} trip makers can be selected from Q

Figure 3.20 SYSTEM MACROSTATES



SOURCE: Putman (1978). Reprinted with permission.

in $Q!/[V_{11}!(Q - V_{11})!]$ ways, according to the familiar combinatorial formula of statistics. Now if we ask in how many ways it is possible to select V_{12} out of the remaining $(Q - V_{11})$ travelers in each case, it is given by $(Q - V_{11})!/[V_{12}!(Q - V_{11} - V_{12})!]$. The total number of ways of selecting V_{11} out of Q *and* V_{12} out of $(Q - V_{11})$ is given by the product of the two combinatorial formulas, or $Q!/[V_{11}!V_{12}![(Q - V_{11} - V_{12})!]$. Continuing on in this way, we see that the total number of ways in which we can select a particular distribution $\mathbf{V} = [V_{ij}]$ distribution from Q is

$$\frac{Q!}{V_{11}!V_{12}!V_{13}!} \quad \text{or} \quad \frac{Q!}{\Pi_{ij}V_{ij}!} \tag{3.52}$$

The combinatorial formula above results regardless of the order in which the entries in **V** are considered. In other words, it is independent of the way we label the zones.

Applying this formula, the number of microstates of mesostate A in Figure 3.18 is $6!/[(V_{11}!)(V_{12}!)(V_{13}!)]$ or $6!/[(3!)(2!)(1!)] = 60$. The number of microstates of mesostate B is $6!/\{(2!)(2!)(2!)] = 90$. By trial and error, one may substitute values for V_{11}, V_{12} and V_{13} in the denominator of the equation and discover that the minimum value of the denominator (subject to the constraint that the sum of all the trips equal six) is at 2!2!2!. This corresponds to the maximum number of microstates 90, which suggests that in the absence of any further information about our example (such as the number of trips originating at a zone and terminating at a zone), the most probable mesostate is when the six trips are evenly distributed to the three destinations. In general, the most probable mesostate is when the number of microstates is to be maximized or Max [$Q!/\Pi_{ij}V_{ij}$]. Stirling's approximation for large values of x yields $\ln(x!) = x\ \ln(x) - x$. Applying such approximation to the logarithm of the above maximization expression results in Equation 3.51, remembering that $Q = \Sigma_{ij}Q_{ij}$ in this case. Thus entropy maximization is shown to be equivalent to the minimum information principle.

XIII. QUALITY OF A MODEL CALIBRATION

In this chapter, we have discussed the various ways to describe the scenario under analysis. We call this descriptive modeling. After all the above work has been performed, the final question arises as to how good the model is in replicating the real world. Obviously, the answers vary depending on whom you ask, and most importantly, the end use of the model. However, here are some scientific measures of merit, which form part of the information on the quality of the model calibration.

A. Chi-Square Test

The chi-square test, for example, can be used to determine how well theoretical probability distributions (such as an assumed normal distribution) fit empirical distributions (in other words, those obtained from sampled data). In general, the chi statistic measures the discrepancy between the estimated and observed frequencies. It is used to test whether a set of estimated frequencies differ from a set of observed frequencies sufficiently to reject the hypothesis under which the expected frequencies were obtained. The formula generally used for the chi-square statistic is,

$$\chi^2 = \sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{\hat{y}_i}$$

where y_i is the observed data and $\hat{y}_i$ is the estimated. If there is a close agreement between the estimated and observed frequencies, χ^2 will be small. If the agreement is poor, χ^2 will be large. A numerical example has been worked out in Section XI-A in connection with the singly constrained gravity model.

Using the chi-square idea, it can be shown that the maximum entropy explanation of spatial distribution can be used to calibrate the parameters such as α's in Equation 3.30. This can be accomplished by the **minimum discrimination information statistic** $2V\Sigma_i\Sigma_j\ V_{ij}\ln(V_{ij}/Q_{ij})$, which has an asymptotic chi-square distribution with $(n'J - L')$ degrees of freedom. Here, L' is the total number of control totals placed on the number of trips made plus the number of parameters to be calibrated (Oppenheim 1995). For example, when the given data D' consist of the number of origin trips V_i, the number of destination trips V_j, (or the equivalent statement about the total number of trips in the study area V), and the total travel cost in trip minutes or trip miles, $L' = n' + |J| + 1$ corresponding to the number of origin zones, the number of destination zones and the b coefficient for the trip distribution curve $\exp(-bC_{ij})$, as shown in Section XI-B. The statistic takes on a value of zero when the model is perfect, in other words, when all predictions V_{ij} are equal to the corresponding observations Q_{ij}, and a positive value for less than perfect model fit. The statistical significance of the model performance may then be tested by comparing the value of the statistic with the threshold value from the chi-square table with appropriate number of degrees-of-freedom at the chosen level of confidence. If the former is greater than the latter, the hypothesis that the distribution of predicted values is not significantly different from that of the observed values must then be rejected. In other words, the calibration needs to be further refined to effect closer agreement between the predicted and observed trips.

B. Variance Reduction

Irrespective of the calibration techniques used, a common measure to compare an estimated model with the observed model is the sum of squared errors:

$$\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{3.53}$$

This measure can be normalized by the number of observations n (where n is a large number), turning it to what is often referred to as the **residual variance:**

$$\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Similarly, the square root can be taken to further normalize the measure

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

which is sometimes called the **standard error of estimate**.

All the above are absolute measures. Depending on the units used in the variables Y, the figures obtained from these formulas will be different. To truly normalize to a relative scale, we first define the worst case as the variance about the mean:

$$\sqrt{\frac{1}{n}\sum\nolimits_i (y_i - \overline{Y})^2}$$

The ratio of Equation 5.53 and the above variance is then a more workable measure of the relative size of the actual variance. We call this ratio ρ. In other words

$$\hat{\rho}^2 = \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{Y})^2}$$

The variance reduction due to the model is simply $1 - \hat{\rho}^2$ The more variance the model can account for, the better. For this reason, it is preferable to have the above equation approaching unity.

The reader will detect the parallel concepts in linear regression, although the above development is, in our opinion, more general and goes well beyond linear models. Thus the model can be calibrated by any calibration technique, particularly when a combination of techniques are used, the measure of merit still readily applies to the combined model, while more specialized measures are only good for each individual model.

Example
As an example, consider the nonlinear doubly constrained gravity model in Section XI-B discussed previously. Here, the $\hat{\rho}^2$ is

$$\frac{\sum_{i=1}^{3}\sum_{j=1}^{3} (y_{ij} - \hat{y}_{ij})^2}{\sum_{i=1}^{3}\sum_{j=1}^{3} (y_{ij} - \bar{\mathrm{Y}})^2}$$

Based on the data contained in Table 3.14 and Table 3.15, we have the relative size of the estimation variance $\hat{\rho}^2 = 3.843 \times 10^{-4}$. The variance reduced by the model is thus $1 - \hat{\rho}^2 = 0.9996$, which is quite impressive. ■

XIV. CONCLUDING REMARKS

We have provided in this chapter a summary of pertinent analysis tools that are useful in describing the context of a facility location/land use decision. The methods are diverse, but they are all spatial extensions of simulation, probabilistic models, and statistical analysis. We also introduce specialized techniques used to analyze spatial interactions, as exemplified by the gravity model, information, and entropy theories. To the extent that warehouses are built to store the appropriate amount in anticipation of deliveries to potential demand points, we review the marginal analysis that governs inventory control over a network. This will help to locate facilities and to make timely delivery of goods and services. Together with the basic building blocks reviewed in appendices to this book, we will have a self-contained set of background tools for the reader. The discussions in these methodological chapters differ from that in the appendices in terms of context. While the appendices are purely applied mathematics, the treatment here is applications-oriented model building. Thus examples are drawn from the subject matter of this book, even though they have to be simplified to achieve the transparency desired. In later chapters, in the CD/DVD software, and in Chan (2005), case studies will be presented wherein we show how the more fully developed models are used to come up with real world decisions.

XV. EXERCISES

Self-Instructional Module: PROBABILITY DISTRIBUTION AND QUEUING

(to be found on the attached CD/DVD)[10]

One of the fundamental insights of the physical and social sciences in the 20th century is the applicability of probability theory. For example, the probabilism of quantum mechanics has replaced the determinism of Newtonian mechanics. Present day analysts speak of deterministic models versus probabilistic models. This module, entitled "Probability Distribution and Queuing," is a continuation of the module on probability. After working through this module, the reader should

- **(a)** understand the concepts of probability distributions
- **(b)** see the applicability of probability distributions in a system of queues.
- **(c)** gain some insights on using queuing theory as a decision-making tool.

This module is fundamental in understanding the topics in Chapter 3, entitled "Descriptive Analysis." In this chapter, the author presents analysis tools such as simulation, subjective probability, econometrics, curve fitting, and information theory. All of these are shown as analytics of spatial information technology. The "Probability Distribution" module also serves as an excellent introduction to the Appendices entitled "Review of Statistical Tools" and "Review of Markovian Processes."

Problem 1: Decision Tree

There are some very helpful software packages to perform Bayesian decision analysis. Aside from commercially developed ones, there is free software such as GeNle at http://genie.sis.pitt.edu/. GeNle has extensive graphic features and it is suitable for large problems. As espoused by the software on the attached CD/DVD, the author likes to adopt a "down to earth" philosophy; he prefers a very basic approach, instead of a more elaborate procedure. Please be mindful that we have mainly pedagogy in mind. Admittedly, the resulting software is not as "high tech" as others. This exercise introduces the users to such a computational approach in solving decision-analytic problems. More importantly, it shows how simple decisions can be combined in a complex decision tree.

Refer to the Bayesian decision tree in Figure 3.11. The readers will agree that it is a rather elaborate tree. We wish to break down the tree into its components, and see how we can combine these components together to form the complete tree. We will analyze the decisions represented in this tree with the aid of a software called TreePlan, which is a very popular software for educational use. TreePlan is an Excel add-in available at nominal cost from DigiBuy (http://www.digibuy.com). The advantage of TreePlan is its simplicity. It works on every desktop or laptop computer that uses Microsoft Office Excel. TreePlan automates the standard Bayesian decision-tree calculations, and display the final results graphically as a tree. Should the reader decide against acquiring this software, s/he can still solve this problem by using straight spreadsheet. The only thing missing is the graphics to display the tree.

Using TreePlan or simply Excel, please reproduce the complete decision tree for the "nuclear power-plant" problem as shown in Figure 3.11. Let us carry this out in four steps, as illustrated by the accompanying Figures below. The basic steps will ease you into TreePlan or Excel, and allow you to be familiarized with the software. The real challenge, or the real thought process for this exercise, is to combine all the stepwise decisions into an overall decision, as shown in the final tree in Figure 3.11.

(a) Decision 1. Refer to the decision tree in Figure 3.21, should a power plant be built (without additional sample data)?

(b) Decision 2. Refer to the decision tree in Figure 3.22, decide on "build" or "no-build" if the sample test is positive.

(c) Decision 3. Refer to the decision tree in Figure 3.23, decide on a "build" or "no-build" decision if the sample test is negative?

(d) Decision 4. Refer to the final decision tree in Figure 3.24, combine all the above decisions and decide on whether the sample test should be conducted to begin with?

Figure 3.21 NO SAMPLING

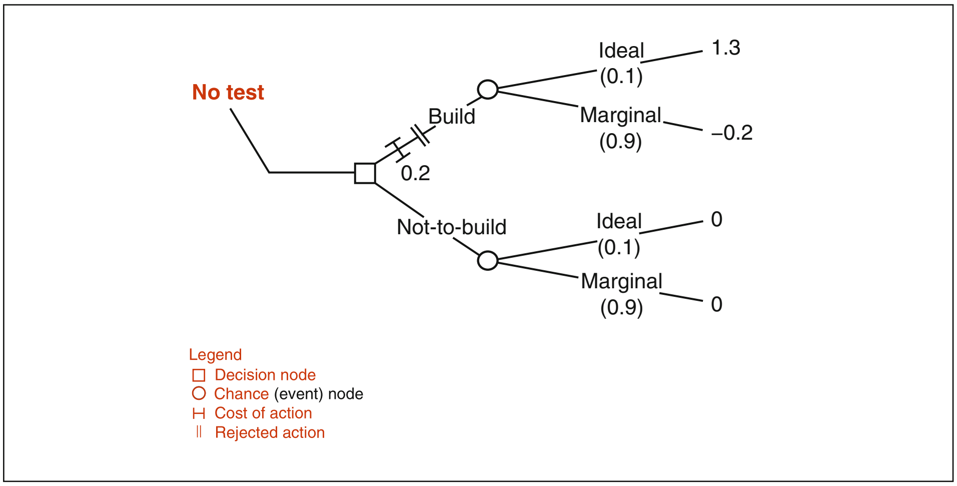


Figure 3.22 DECISION UPON POSITIVE TEST

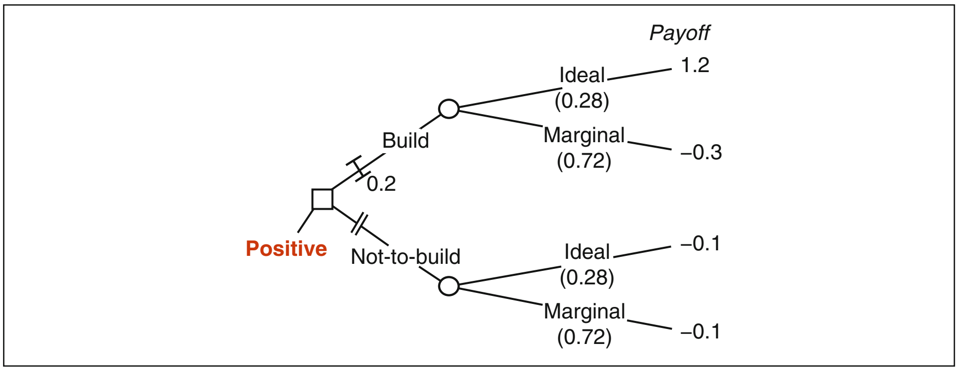


Figure 3.23 DECISION UPON NEGATIVE TEST

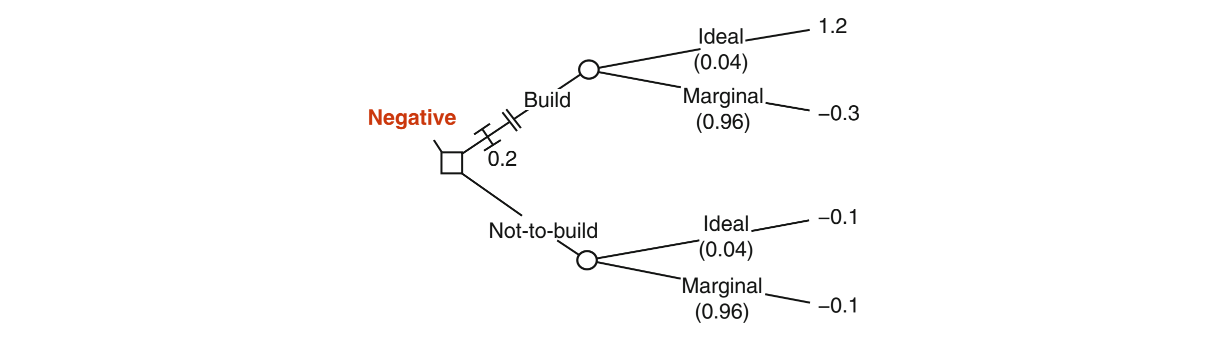
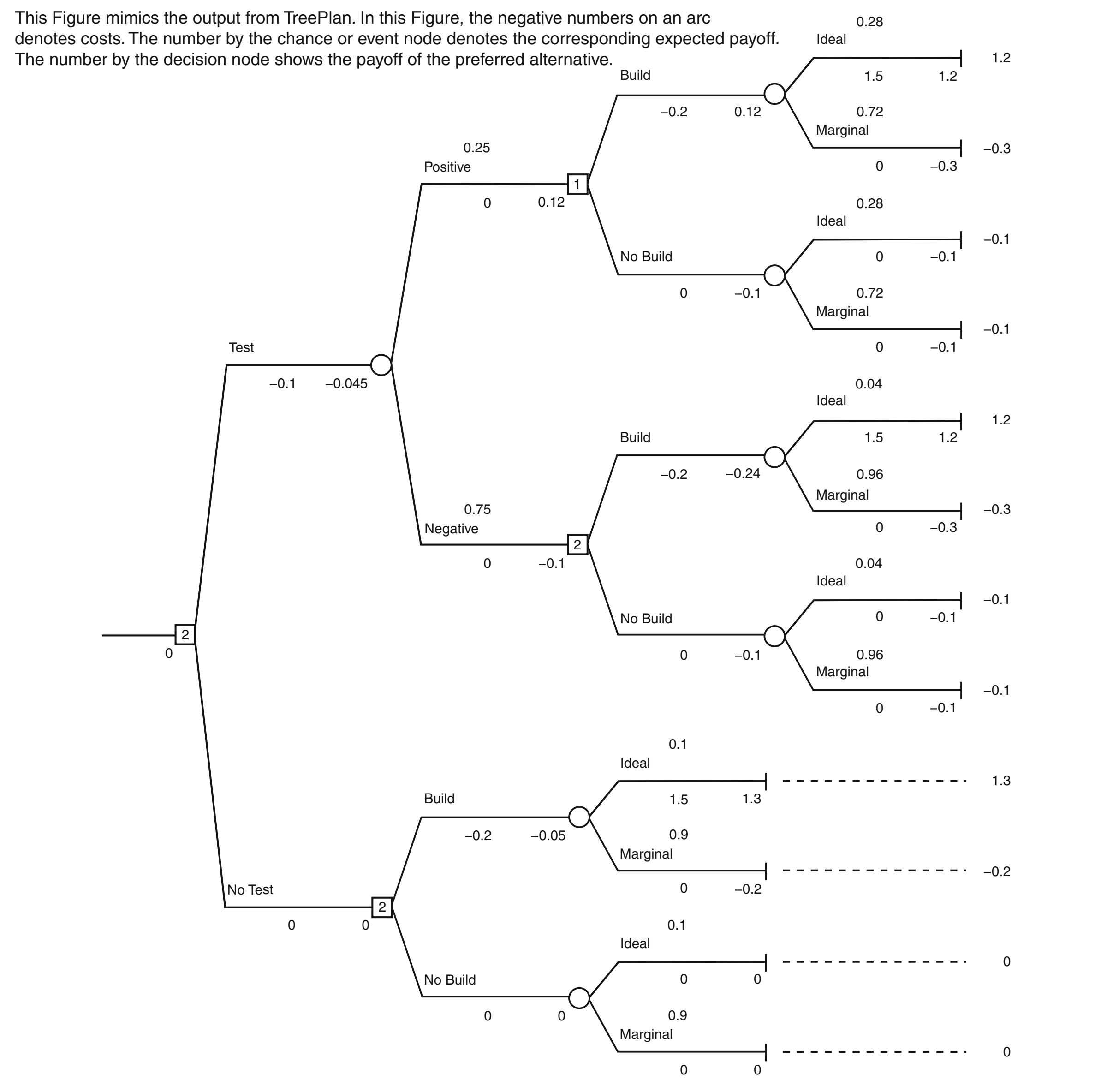


Figure 3.24 DECISION TREE FOR TEST (INCLUDING COSTS)

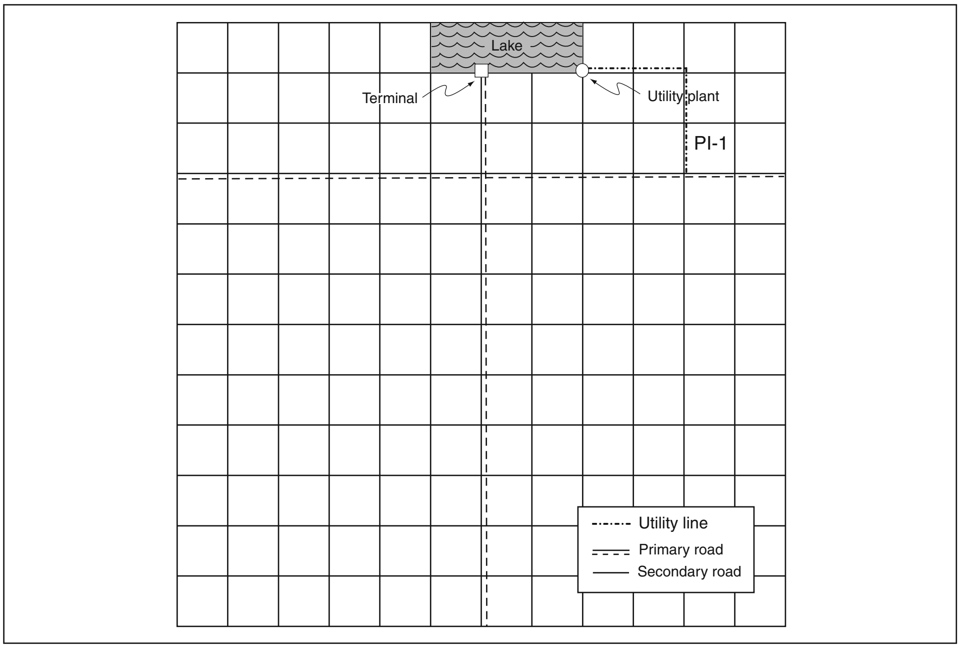

This Figure mimics the output from TreePlan. In this Figure, the negative numbers on an arc denotes costs. The number by the chance or event node denotes the corresponding expected payoff. The number by the decision node shows the payoff of the preferred alternative.

Problem 2: Simulation

The Community Land Use Game (CLUG) is discussed in book Chapter 3, Section IV. The game brings out some of the non-quantifiable elements of urban and regional development—an element not addressed adequately by the analytical models. Parallel to the concept of Economic Base Theory, three economic sectors are represented. The basic sector consists of Full Industry and Partial Industry. The residential sector is made up of Partial Industry. The residential sector is made up of single-unit, double-unit, triple-unit, and quadruple-unit housing: R1, R2, R3, plus R4; and the service sector is exemplified by Local Store plus Central Store. Among the many factors that shape the community development is transportation (as evident from the results of our CLUG discussion)–a fact that can be verified by the following example (reference the CLUG Playing Board Diagram, Figure 3.25).

An entrepreneur is deciding between two sites for his Partial Industry (PI), as marked by PI-1 (6-72) and PI-2 (10-70) on the playing Board. The residential quarters of his labor force are located at R2 (12-8). With the primary road system, transportation terminal and the required utility line already in place and paid for, you may wish to answer the trailing questions by sketching in any remaining infrastructure to be built. An example utility line has already been drawn for you above in the CLUG playing board. (For further explanation of these calculations, please consult book Section 4-ll-A, where a similar example has been worked out.)

Figure 3.25 SAMPLE CLUG PLAYING BOARD



SOURCE: Feldt (1972). Reprinted with permission.

(a) Based on the resulting built infrastructure, which is the preferred site location? Please document in the Table below the calculations you need to arrive at this conclusion. Remember that secondary roads carry a cost of two units, while primary roads carry a cost of one unit.

Site PI-1	**Site PI-2**
Transportation cost from R2 to PI-1:	Transportation cost from R2 to PI-2:
Transpo cost to export goods via the Port:	Transpo cost to export goods via the Port:
Total transpo cost:	Total transpo cost:

(b) Now explain in words the reasons behind your choice of the Partial Industry location.

From the point of view of the community, who has to pay for the construction of the utility lines, a different financial analysis is necessary. For our purposes, we can assume that a land parcel for R2 and another for PI are committed and purchased in Round 1 of the game, with the corresponding utility line constructed in the same Round (please draw in the utility lines in the CLUG board shown as book Figure 3.25 above). All buildings are constructed and fully operational in Round 2.

(c) Now execute the 11-step development sequence as described in book Section 3-IV. Please organize your calculations using the attached Financial Status Table (Figure 3. 26, which has to be completed in full). Notice that a parcel has to be serviced by a utility line on at least one of its four faces before construction can take place. The construction and land costs are given in a table below. Notice that assessed value is 50 percent of construction and land costs, and tax

Figure 3.26 CLUG COMMUNITY FINANCIAL STATUS

Clug Community Financial Status

	Site	Total Community Assessed Value	Utility Costs @$2,000/New @$1,000/Old		Social Service Costs @ $1,000 Per Residential Unit Cost		Debt Service @ 10% Previous Deficit	Total Community Costs	Total Taxes Raised	Round Surplus or Deficit	Cummulative Surplus or Deficit	Debt Limit @ 10% of Total Assessed Value	Debt As % of Debt Limit
			No. New	Cost	No. Units								
			No. Old	Cost									
Round 1	PI-1												
	PI-2												
Round 2	PI-1												
	PI-2												

is five percent of assessed value of property. (For the purpose of this problem, you can assume no deterioration on buildings.)

Unit characteristics (in $1000's)	Construction	Land cost
F1	$96	10
P1	$48	10
LS	$24	10
CS	$24	10
O	$36	10
R1	$12	5
R2	$30	5
R3	$48	5
R4	$72	5

(d) Please decide, from the calculations up to the Second Round, which Partial Industry is preferred. Why?

ENDNOTES

[1] A numerical example is worked out in Chapter 4 for illustration.

[2] A methodological review of stochastic process may be found in Appendix 3.

[3] For an explanation of the *t*-statistic, see Appendix 2. Paired *t*-tests are used here, which test the hypothesis that there is no significance between two sample means, m_1 and m_2, or the difference between the sample means is zero. $m_d = m_1 - m_2 = 0$. Traditional statistical tests are characterized by significance levels, or the confidence one would place on the test results.

[4] A better term is the "News vendor" problem.

[5] The correlation coefficient is defined in Appendix 2.

[6] The matrix notation is adopted here for ease of explanation only, it is not essential for development. For a formal introduction to the matrix representation of linear regression, refer to Appendix 2.

[7] For a discussion of the statistical and practical considerations in selecting a regression equation, see Appendix 2. A parsimonious model has the proper balance between practicality and statistical significance.

[8] This is referred to as constrained regression.

[9] Entropy originates from the Greek and roughly means change or transformation.

[10] The answer to this Module is attached at the end of this textbook.

REFERENCES

Ahituv, N.; Berman, O. (1988). *Operations management of distributed service networks: A practical approach.* New York: Plenum Press.

Al-Mosaind, M. A.; Ducker, K. J.; Strathman, J. G. (1993). "Light rail transit stations and property values. A hedonic price approach." (Presentation Paper 930806). Paper presented at the 72nd meeting of the Transportation Research Board, Washington D. C.

Black, W. R. (1991). "A note on the use of correlation coefficients for assessing goodness-of-fit in spatial interaction models." *Transportation* 18:199–206.

Cesario, F. J. (1975). "A primer on entropy modeling." *AIP Journal* (January):40–48.

Chan, Y. (2005). *Location, transport and land-use: Modeling spatial-temporal information.* New York: Springer-Verlag.

Claunch, E.; Goehring, S.; Chan, Y. (1992). Airport location problem: Three and four city cases. Working Paper. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson AFB, Ohio.

Clemen, R. T. (1996). *Making hard decisions: An Introduction to decision analysis*, 2nd ed. Pacific Grove, California: Duxbury Press.

Cliff, A. D.; Ord, J. K. (1975). "Space-time modeling with an application to regional forecasting." *Transactions—Institute of British Geographers* 66:119–128.

Congressional Office of Technology Assessment (1982). *Global models, world futures, and public policy.* Washington, D. C.: U. S. Congress.

De Neufville, R: Stafford, J. (1971). *Systems analysis for engineers and managers.* New York: McGraw-Hill.

Dickey, J. W. (1983). *Metropolitan transportation planning,* 2nd ed. New York: McGraw-Hill.

Duann, L-S.; Chang, C-C. (1992). "A study of the development of disaggregate residential location choice models." *Transportation Planning Journal* (Taiwan, Republic of China) 21, No. 4:401–422.

Fanueff, R.; Sterle, T.; Chan, Y. (1992). A warehouse location, inventory allocation problem. Working Paper. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson AFB, Ohio.

Feldt, A. G. (1972). CLUG: *Community Land Use Game.* New York: Free Press.

Foot, D. (1981). *Operational urban models: An introduction.* New York: Methuen.

Golan, A. (1998). "Maximum entropy likelihood and uncertainty: A comparison." In *Maximum entropy and Bayesian methods,* edited by G. J. Erikson. Boston: Kluwer Academic Publishers.

Gonzalez, R. C.; Woods, R. E. (1992). *Digital image processing.* Reading, Massachusetts: Addison-Wesley.

Guy, C. M. (1991). "Spatial interaction modelling in retail planning practice: the need for robust statistical methods." *Environment and Planning B* 18: 191–203.

Heller, M.; Cohon, J. L.; Revelle, C. S. (1989). "The use of simulation in validating a multi-objective EMS location model." *Annals of Operations Research* 18: 303–322.

Hurter, A. P.; Martinich, J. S. (1989). *Facility location and the theory of production.* Boston: Kluwer Academic Publishers.

Kalaba, R.; Moore, J. E.; Xu, R.; Chen, G. (1995). A new perspective on calibrating spatial interaction models: An application to shopping centers. Working Paper. School of Urban and Regional Planning. University of Southern California. Los Angeles.

Kanafani, A. (1983). *Transportation demand analysis.* New York: McGraw-Hill.

Lapin, L. (1975). *Quantitative methods for business decisions with cases,* 4th ed. Orlando, Florida: Harcourt Brace Jovanovich.

Lee, S-Y.; Haghani, A. E.; Byun, J. H. (1995). Simultaneous determination of land use and travel demand with congestion: A system dynamics modeling approach. (Presentation Paper 950716). Paper presented at the 74th meeting of the Transportation Research Board, Washington D. C.

Meadows, D. H.; Meadows, D. L.; Randers, J.; Behrens, W. W., III (1972). *The limits to growth.* New York: Signet.

Oppenheim, N. (1980). *Applied urban and regional models.* Englewood Cliffs, New Jersey: Prentice Hall.

Oppenheim, N. (1995). *Urban travel: Demand modeling.* New York: Wiley-Interscience.

Pritsker, A. B. (1986). *Introduction to simulation and SLAM,* 3rd ed. New York: Halsted Press.

Putman, S. (1978). The integrated forecasting of transportation and land use. In *Emerging transportation planning methods,* edited by W. Brown. Washington D. C.: Office of University Research. Research and Special Programs Administration. U. S. Department of Transportation, 119–147.

Repede, J. F.; Bernardo, J. J. (1994). "Developing and validating a decision support system for location emergency medical vehicles in Louisville, Kentucky." *European Journal of Operational Research* 75:567–581.

Rubenstein, B.; Zandi, I. (1998). An evaluative tool for solid waste management. Working Paper. Department of Information Systems. University of Maryland at Baltimore County. Baltimore, Maryland.

Rue, H. (1995). "New loss functions in Bayesian imaging." *Journal of the American Statistical Association* 90:900–908.

Senior, M. S. (1973). "Approaches to residential location modeling 1: Urban ecological and spatial interaction models (a review)." *Environment and Planning* 5:165–197.

Thompson, G. L.; Weller, B.; Terrie, E. W. (1993). New perspectives on highway investment and economic growth. (Presentation Paper 930597). Paper presented at the 72nd meeting of the Transportation Research Board, Washington D. C.

Tsoukias, A. (2008). "From decision theory to decision aiding methodology." *European Journal of Operational Research* 187:138–161.

Webber, M. J. (1984). *Explanation, prediction and planning: The Lowry model.* London: Pion.

Wikipedia (2010). "Influence diagram." Website as last modified on 9 December 2010.

Willemain, T. R. (1981). *Statistical methods for planners.* Cambridge, Massachusetts: MIT Press.

Winston, W. L. (1994). *Operations research: Applications and algorithms,* 3rd ed. Belmont, California: Duxbury Press.

4

Prescriptive Tools for Analysis

"The mathematical sciences particularly exhibit order, symmetry and limitation, and these are the greatest forms of the beautiful."
Aristotle

This chapter will follow our discussion on the two methods of analysis identified in our taxonomy, focusing now on prescriptive instead of descriptive techniques. By way of a definition, prescriptive technique is generally used in system design, where a stated goal or objective is to be achieved. In the context of this book, the function of a prescriptive model then, is to configure a facility location or land use plan to achieve this goal or objective. For example, if one is to stimulate residential development in an area, the model, after it has been set up, will prescribe a land use plan that will provide all the utilities, transportation, and zoning that will best facilitate such a development. To the extent that we often wish to provide the best design, optimization procedures are an integral part of the prescriptive tool kit. Here in this chapter, we will introduce the basic building blocks of prescriptive analysis (including optimization concepts), deferring most of the implementation and computational details to subject focused chapters throughout this book and the appropriate book appendices. Included in the latter category are such appendices as "Optimization Schemes" (Appendix 4) and "Control, Dynamics, and System Stability" (Appendix 1).

I. A TYPICAL PRESCRIPTIVE MODEL

As always, examples are the best way to introduce a new concept. We will build upon the three-sector urban economy example from Chapter 3, namely, an economy made up of the residential, basic, and service employment sectors. Three fundamental steps are involved in effecting a prescriptive model: defining goals and objectives, representing the system, and then putting them together in a single model. Inasmuch as step two calls for system representation, clearly prescriptive techniques are not mutually exclusive of descriptive techniques. A prescriptive

Y. Chan, *Location Theory and Decision Analysis*, 2nd ed., DOI 10.1007/978-3-642-15663-2_4,

model can be thought of as an extension of a descriptive model, in which goals and objectives are added on top.

A. Goals and Objectives

First, let us discuss the goal or the objective one tries to optimize, which is an important part of a prescriptive model. Two examples of the objective function may be cited from Chapter 2: efficiency and equity. Efficiency may mean the least costly way to provide quality housing, while equity may be concerned with constructing housing in such a manner that it is equally accessible to all the population. Oftentimes, the objective function is also referred to as the **figure of merit.** Thus the efficiency figure of merit in the example is the total cost, which is to be minimized. Accessibility may become the equity measure for the example, which is to be maximized. One way to do this may be to formulate a land use/transportation plan that would guarantee accessibility to housing within 15 minutes of travel time from work for all the urban population. The question now becomes: Can we configure a land use plan that will achieve both efficiency and equity?

Another feature of a prescriptive model can be cited. Suppose one has a parcel of land on which he or she wishes to build housing or offices in order to maximize the utility of the land but there are a number of encumbrances, one of which may be the development density allowable by zoning codes. Thus in an area zoned for single family units, the parcel cannot be developed into a multiple dwelling apartment building no matter how much the developer wants it to be. Thus a prescriptive model has to include the representation of the scenario under which one operates, including the many constraints such as zoning density. We have already seen in Chapter 3 how these constraints can be modeled. To the extent that descriptive techniques are adept in system representation, it is relatively straightforward to build a prescriptive model on top of a descriptive model by simply adding objective functions. We will illustrate how this is performed.

B. Representation of the System

The scenario under study and the interrelationship between all the components are first represented in a flow chart or a set of simultaneous equations. These equations are formulated in a manner similar to those introduced in the descriptive modeling discussion. The simulation approach, for example, requires a flow chart to represent the system. The econometric approach, on the other hand, is formulated in a set of simultaneous equations. Again, some examples may be useful. A flow chart of a basic economy may include the relationship between manufacturing, retail, and household sectors in an algorithmic set of steps, as shown in the economic-base example in previous chapters. A correlative model, on the other hand, may start with an arrow diagram showing primary, secondary, and tertiary relationships among all the variables. The relationships are then formalized into a set of simultaneous equations, with the coefficients a, b, c, and d to be calibrated by econometric techniques. We review an example initiated in Chapter 3 below:

$$\begin{aligned}&\textit{(forecast pop)} = a\ \textit{(forecast emp)} + b\ \textit{(base-yr pop)}\\&\textit{(forecast emp)} = c\ \textit{(forecast pop)} + d\ \textit{(base-yr emp)}\end{aligned}\qquad(4.1)$$

Here the coupling effects between population and employment are explicitly recognized, in that employment needs to be supported by a labor force, and at the same time, dependent population often follows employment.

C. A Prescriptive Formulation of the Economic-Base Concept

Now we can show how the familiar descriptive formulation of the economic-base theory can be converted to a prescriptive format by the introduction of objective functions. The economic-base model deals with four types of land use: retail, residential, manufacturing, and undeveloped, all of which are to be placed within the available land. In accordance with economic-base theory, basic manufacturing land is exogenously determined and hence a constant. Undevelopable land is the same way. This leaves us with two decision variables, retail land use and residential land use. All the development has to be contained within the available land. Our example problem then can be represented by the first of several constraint equations, where, once again, the retail and residential land use are decision variables, while manufacturing and undevelopable land are treated as constants.

$$\textit{(retail land use)} + \textit{(residential land use)} \leq \textit{(developable land)}$$

Another constraint deals with density and zoning, that the maximum development density as permitted by zoning cannot be exceeded:

$$\textit{(no. of dwelling units)/(res. land use)} \leq \textit{(Max allowable density)} \tag{4.2}$$

The last constraint is worthy of mention. It models the often observed fact that in order to establish any retail activity in a zone, one shall have a minimum threshold of viable activities, often referred to as the critical mass, or the smallest amount of activities that can sustain the business:

$$\textit{(retail emp density)(retail land use)} \geq \textit{(threshold)} \tag{4.3}$$

Now if one adds an objective function on top of the set of equations, which represent the system, a prescriptive model is obtained. Instead of the efficiency and equity objectives, a plausible objective may be to maximize the development of the land, which may be measured in terms of the total employment and population in the area:

$$\text{Max}[\textit{(retail emp density)(retail land)} + \textit{(res density)(res land)}] \tag{4.4}$$

As can be seen, a distinguishing feature of the example model is that it consists of a set of simultaneous equations. In real life, it may not be possible to model the system as analytically as shown here. It is entirely conceivable that simulations is the only means to represent the complex operations of the system under discussion. Still, objective functions can be imposed on top to effect a prescriptive model in this situation.

II. HEURISTIC SOLUTION TECHNIQUES

Having fixed some fundamental ideas about a prescriptive model, let us proceed to survey some of the tools for solving such a model. We have categorized prescriptive tools into two classes. One is heuristic and the other analytical. **Heuristic techniques** refer to a set of methodologies that are not strictly mathematical. They may consist merely of a number of clever or intuitive computational procedures. **Analytical techniques,** on the other hand, are more mathematically rigorous and have a transparent or traceable relationship between the constituent components. Heuristic techniques consist basically of a set of carefully specified computational schemes that are usually programmed into the computer to yield good solutions. These techniques are devoid of the nice, transparent properties that characterize analytical procedures, and an optimal solution is not often guaranteed. Three types of heuristic techniques will be discussed here: the manual approach, the enumerative, and the direct search.

A. Manual Approach

The **manual approach** is a very simple approach. It involves formulating alternative plans that represent the various zoning and transportation policies, for instance, then performing the forecast as a second step, and finally picking the plan that yields the best figure of merit. Another example would be to pick two candidate locations for a facility, evaluate the merits of both, and pick the better of the two. Take the Community Land Use Game (CLUG) discussed in Chapter 3. Three economic sectors parallel to economic-base theory are represented: the basic sector consisting of full industries and partial plants, the residential sector consisting of R1, R2, R3, and R4 housing (in order of higher development-density), and the service sector as exemplified by Central Store (CS) and Local Store (LS).

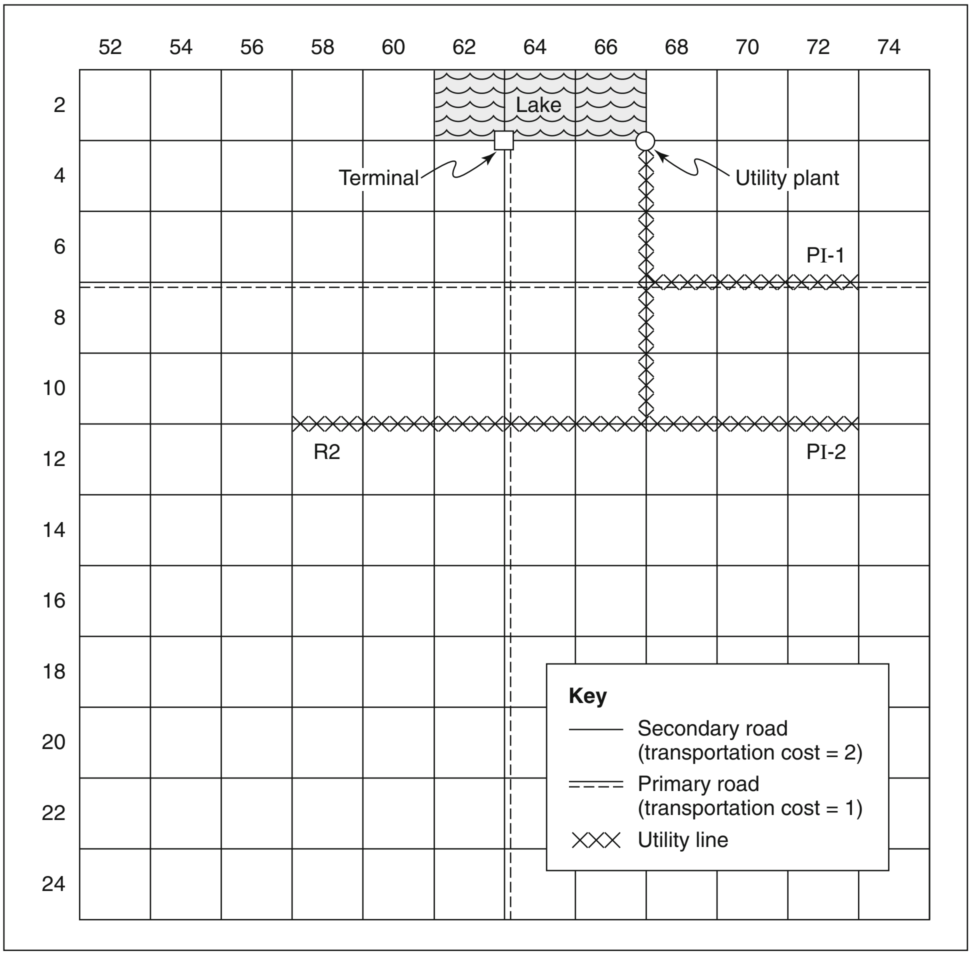
Referring to Figure 4.1, among the many factors that shape the community development is accessibility to the export market through the harbor terminal. Let us say an entrepreneur is deciding between two sites for his partial industry (PI), as marked by PI-1 (grid point 6-72) and PI-2 (grid point 12-72) on the playing board shown in Figure 4.1. The residential quarters of his labor force is located at R2 (12-58). With the given road system, transportation terminal, and the required utility line already in place and paid for, which is the preferred site location judging purely on transportation cost?

Site PI-1: accessibility to population = (6)(1) + (2)(2) = 10
accessibility to export market = (6)(1) = 6
total transportation cost = 16;

Site PI-2: access to population = (6)(2) = 12
access to market = (6)(1) + (2)(2) = 10
total transport cost = 22

Based on the above manual analysis, PI-1 is the best choice because it has a lower transportation cost altogether.

Figure 4.1 LOCATIONAL CHOICE USING A MANUAL PRESCRIPTIVE TECHNIQUE



B. Enumerative Method

The **enumerative method** is often used when the options can be represented in discrete integer variables. For example, four divisions of a company *A*, *B*, *C*, and *D* have a choice among four sites, 1, 2, 3, and 4, to locate a plant. A prescriptive model is used to assign each division to a site according to some figure of merit such as overall cost to the company. The integer variable x_{ij} assumes the value of 1 when industrial plant *i* is located at site *j*. Consider the costs associated with locating four plants in four sites as shown in Table 4.1. The assignment of a plant to a site would be made according to the lowest total cost, where each site can take only one plant and no more. Such an assignment can be computed after a combinatorial programming model, such as the following, has been formulated:

Table 4.1 SITE LOCATION COST OF INDUSTRIAL PLANTS

		Sites			
		1	2	3	4
Plants	*A*	9	5	4	5
	B	4	3	5	6
	C	3	1	3	2
	D	2	4	2	6

$$\begin{aligned} &\text{Min} \sum_{i=1}^{4} \sum_{j=1}^{4} c_{ij} x_{ij} \\ \text{s.t.} &\sum_{j=1}^{4} x_{ij} = 1 \qquad i = A, B, C, D \\ &\sum_{i=1}^{4} x_{ij} = 1 \qquad j = 1, 2, 3, 4 \\ &x_{ij} = \{0, 1\} \end{aligned} \tag{4.5}$$

where the c_{ij}'s are defined in the cost table.

In long hand, the model can be spelled out. The objective function now becomes:

$$9x_{A1} + 5x_{A2} + 4x_{A3} + 5x_{A4} + 4x_{B1} + 3x_{B2} + \cdots$$

The constraints become:

$$\begin{aligned} x_{A1} + x_{A2} + x_{A3} + x_{A4} &= 1 \\ x_{B1} + x_{B2} + x_{B3} + x_{B4} &= 1 \\ &\cdots \\ x_{A1} + x_{B1} + x_{C1} + x_{D1} &= 1 \\ x_{A2} + x_{B2} + x_{C2} + x_{D2} &= 1 \\ &\cdots \end{aligned}$$

While there are more efficient solution methods, such a problem can be solved by an enumerative scheme such as **branch and bound** (B&B). Here we describe a general B&B procedure (Hillier and Lieberman 1990). The algorithm is described for both maximization and minimization problems, with the former described in the general text and the latter in parenthesis:

Step 0: *Initialization.* $z_L(z_U)$ = value of best known feasible solution. (If none, $z_L[z_U] = -\infty\ [+\infty]$.) Go to Step 2.

Step 1: *Branch.* Based on some rule, select unfathomed node and partition it into two or more subsets (subproblems/nodes).

Step 2: *Bound.* For each new subset (subproblem/node), find an upper (lower) bound $z_U^i(z_L^i)$, for example, by solving a relaxed subproblem for the objective function value of feasible solutions in the subset.

Step 3: *Fathom.* For each new subset *i*, exclude *i* from further explicit enumeration if

a) $z_U^i(z_L^i) \leq (\geq) z_L(z_U)$;

b) Subset *i* cannot have any feasible solutions; and

c) Subset *i* has a feasible solution. If $z_U^i(z_L^i) > (<) z_L(z_U)$, set $z_L(z_U) = z_U^i(z_L^i)$ and store as the incumbent solution.

Step 4: *Stopping rule.* Reapply test (a) to all live (unfathomed) nodes. If no unfathomed nodes remain, stop. Incumbent solution is optimal. Else, return to step 1.

Alternatively, let $z_L^*(z_U^*) = \text{Max}_i\, z_L^i\, (\text{Min}_i\, z_U^i)$. Stop when $z_L(z_U)$ is within ϵ percent of optimal solution.

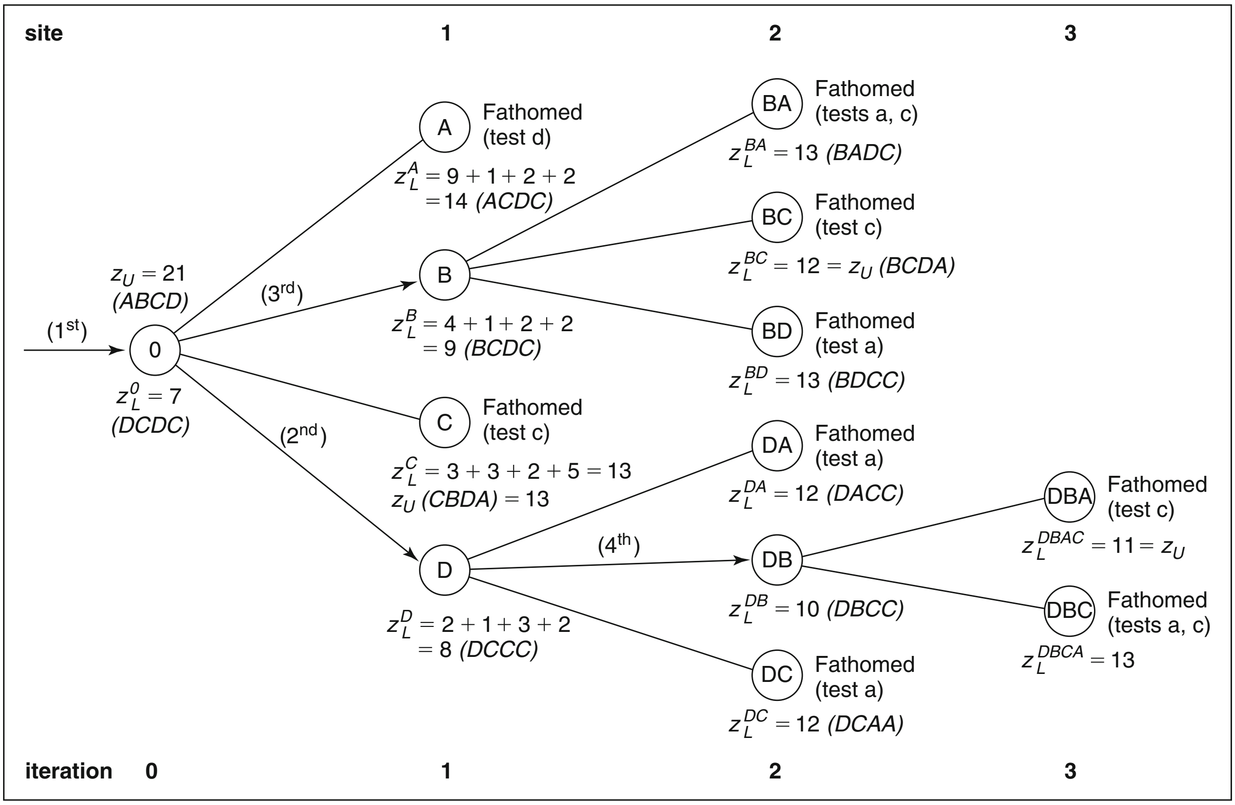
There are two problem specific rules that need to be supplied to the general algorithm (Hillier and Lieberman 1986):

Branch. Look at all possible ways of assigning next plant to unassigned site. Use best bound.

Bound. At any node, add lowest cost assignment to current solution whether feasible or not, in other words, for all unassigned industries, assign the lowest cost site. The only exception is where assignments have already been made, then the plant cannot be assigned a second (additional) site.

The B&B tree is shown in Figure 4.2. At each node of the tree, lower and upper bounds z_L and z_U need to be computed. In the initial node 0, for example, the lower bound is simply to assign plants to sites irrespective of the rule that says "one plant, one site." Thus the popular plants have the choice of more than one site: for example, plant *D* can locate in both sites 1 and 3, *C* in both 2 and 4. On the

Figure 4.2 BRANCH AND BOUND TREE



SOURCE: Hillier and Lieberman (1986). Reprinted with permission.

Table 4.2 MODIFIED SITE LOCATION COST DURING BRANCH AND BOUND

		Sites			
		1	2	3	4
Plant	*A*	(~~9~~)	~~5~~	~~4~~	~~5~~
	B	~~4~~	3	5	6
	C	~~3~~	(1)	3	(2)
	D	~~2~~	4	(2)	6

other hand, if the one-plant-one-site rule is followed, an easy way to assign is to have plant *A* assigned to site 1, *B* to 2, *C* to 3, and *D* to 4. While the former way of assigning will achieve an overall cost lower than reality, the latter will certainly be more costly than necessary; hence the former and latter constitute the lower and upper bounds respectively. The upper and lower bounds define the range within which the final solution will reside.

At node *A* of iteration 1, the lower bound can be evaluated by working with the cost table by striking out the row and column denoting the commitment of a plant to a site. In Table 4.2, for example, row *A* and column 1 are struck out. With the remaining cells in the table, a lowest cost assignment is again obtained, irrespective of whether it is a feasible assignment or not; in other words, whether or not the "one plant, one site" rule has been violated. Hence the lower bound is to assign plant *C* to site 2, *D* to 3, and again *C* to 4 (inasmuch as *C* is popular). We call this solving the relaxed subproblem. Node *B* in iteration 1 is evaluated similarly.

At node *C*, a feasible solution is obtained. According to Step 3(c) of the B&B algorithm, the node has been fathomed in the sense that an incumbent solution has been obtained. Also, a new upper bound is obtained by setting $z_U = z_L = 13$. Intuitively, it says that we will not accept solutions that are worse (bigger) than 13 in the objective function value from this point on, inasmuch as we already have an incumbent solution with this overall cost. Hence node *A* is now eliminated from further consideration by way of fathoming rule 3(a), which says, "Prune the branch that has a overall cost bigger than the current upper bound."

The algorithm proceeds until all possible assignment combinations have been implicitly enumerated through the fathoming rules. The iterations can be summarized by the following table, which documents the steady improvement of the incumbent solution in terms of a lower total cost:

Iteration	z_U	z_L	Assignment
0	21	7	*ABCD*
1	13	8	*CBDA*
2	12	10	*BCDA*
3	11	11	*DBAC*

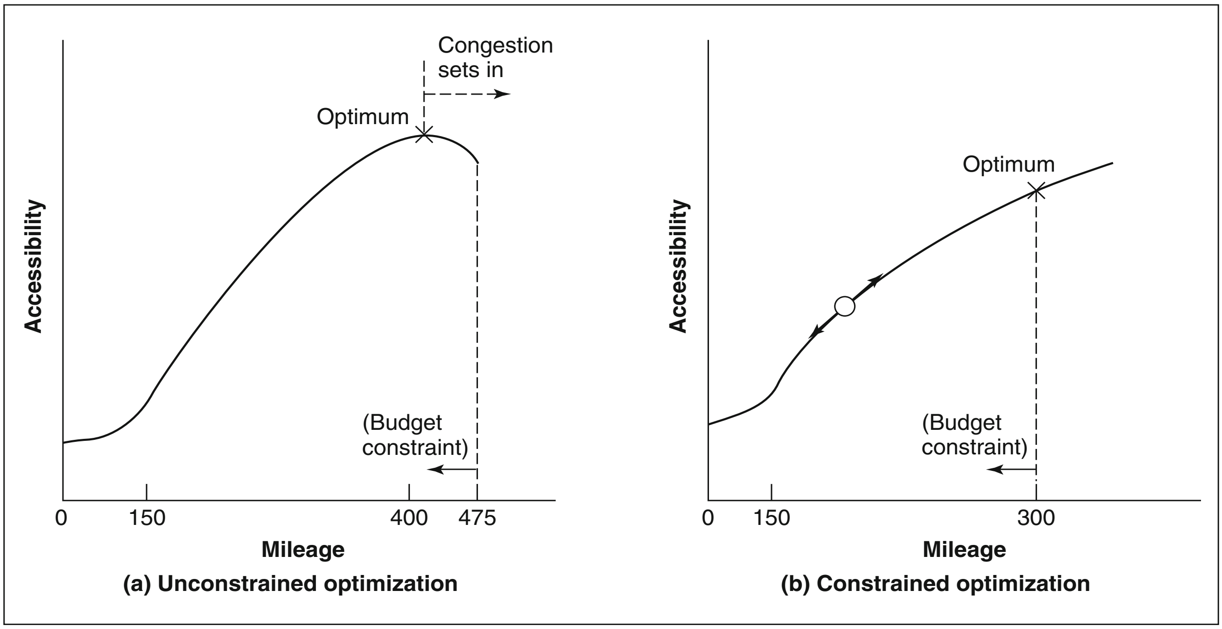
This table can be referenced against Figure 4.2. The iterations refer to figure columns for z_U amd branching sequence for z_L. The optimal solution, shown at node *DBA* in iteration 3, is $x_{A3} = x_{B2} = x_{C4} = x_{D1} = 1$ with the rest of the decision variables equal to zero. This means plant *A* is to be located in site 3, plant *B* in 2, *C* in 4, and *D* in 1.

C. Direct Search Technique

When the decision variable is continuous rather than discrete, a **direct search technique** is very common in optimization. For example, we wish to build the optimal mileage of highways to obtain the most accessibility for the entire region as a whole. Given the relationship between accessibility and highway miles as shown in Figure 4.3, a direct search procedure can be used to identify the highway miles which provide the best accessibility for a fixed budget. The direct search technique simply explores the shape of the objective function (which is accessibility in this case) experimentally. We have shown two possible shapes of the function in Figure 4.3 depending on whether the problem is constrained by the budget or when congestion sets in at some point. In the latter case, Figure 4.3(a) shows that any additional highway miles built beyond the congestion point will decrease accessibility rather than increase it.

The shape of the function as shown in Figure 4.3(a) is a concave function between 150 and 475 miles of constructed highway. Maximizing a concave function over a convex region such that the mileage ranging from 150 to 475 will yield the unique optimum of 400. This is shown in Part (a) of the figure. On the other hand, if the new addition in highway mileage is limited by the budget to 300 as suggested above, it becomes a constrained optimization problem (figure 4.3(b)). The optimum in this case will be at 300 instead. Maximizing a concave objective function over a convex region—such as the continuous line segment from 150 to 300 or 475—is termed a convex programming problem. Barring special circumstances, uniqueness of the optimum may be guaranteed. Should the range be expanded now to 0–475 in Part (a) of the figure and 0–300 in Part (b) of the figure, the objective function for accessibility is no longer concave, since the function over the range 0 to 150 is convex. The change from a convex to concave function occurs at the inflection point of 150 as shown.

Figure 4.3 A DIRECT SEARCH TECHNIQUE

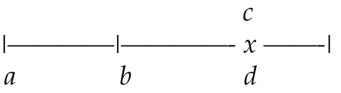


(a) Unconstrained optimization

(b) Constrained optimization

Oftentimes, the shape of the objective function is unknown, although we have a good assurance that it has only one mode. In other words, there is only one maximum point rather than several optimal points, consisting of a global optimum. By **global** or **local optima** we mean the "mountain top" and other "hilltops" respectively. In this case, the Fibonacci search technique will locate the optimum quite efficiently. **Fibonacci search** is one of the most efficient ways to allocate *m* allotted search points for this function. The method consists of computing the value of $g(x)$, the objective function to be optimized, in *m* points. Each of the *m* points is chosen in such a way that having obtained the result for each new point, we can eliminate a subinterval—as large as possible—of the current interval. This process ensures that the optimum will not be located within the subinterval.

A special Fibonacci procedure called the method of golden section will serve to introduce this technique, where *m* is variable, instead of fixed. We wish to locate the optimum in as few trial points as possible. The logic behind the method of Golden Section is based on elimination of the range of search based on the results of existing search points covered. Suppose $g(x)$ is defined between *a* and *d*:



If searches have been conducted in *b* and *c*, due to unimodality and another provision, and $g(b) < g(c)$, one can discard *ab*. The points to search are based on a fixed ratio:

$$\frac{whole}{larger} = \frac{larger}{smaller} = constant = 1.618$$

For example, for the interval *ad* shown above, the search points *b* and *c* are determined by

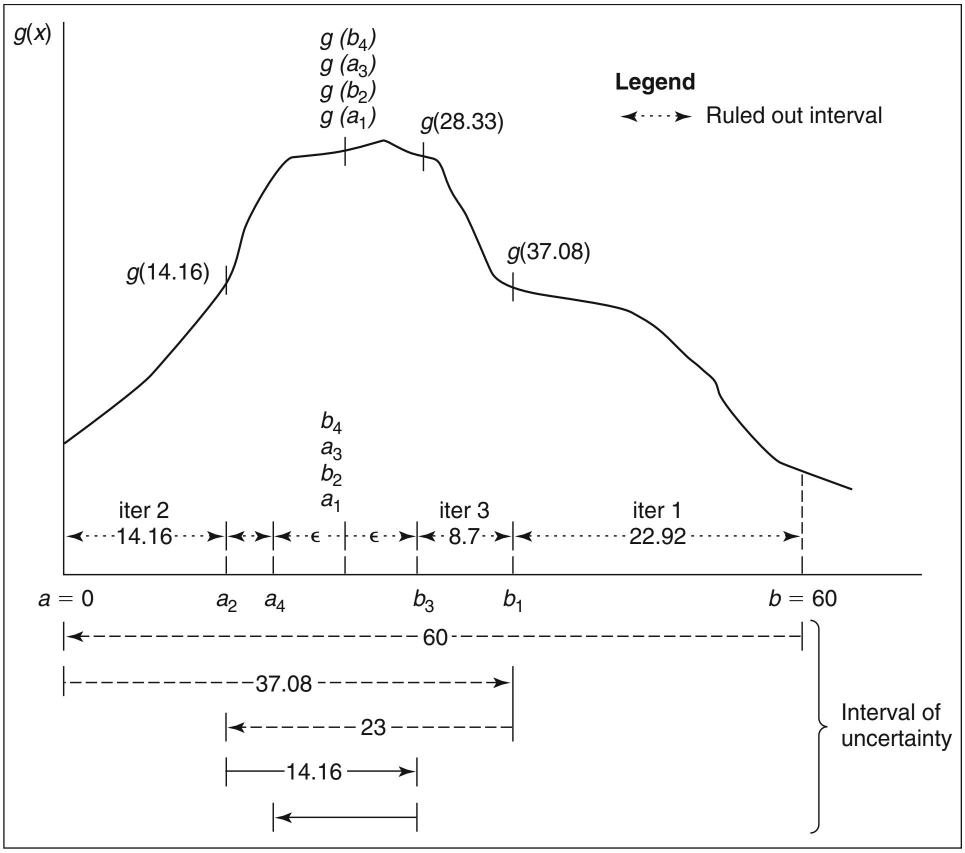
$$\frac{ad}{bd} = \frac{bd}{ab} = 1.618$$

This ratio and its reciprocal 0.618 have a long history of use in design, particularly in architecture.

D. The Golden Section Algorithm

An example will illustrate the **Golden Section algorithm.** Suppose a retailer is to locate a shop to capture as large a market as possible among the competitors. The retailer is considering a stretch of highway 60 miles (96 km) in length, within which the shop is to be located. Such a problem can be solved by the Golden Section method, a special application of Fibonacci search. Throughout the algorithm, we will refer to Figure 4.4, showing an unknown unimodal function representing the market potential along the highway.

Figure 4.4 UNIMODAL FUNCTION WHOSE PRECISE SHAPE IS UNKNOWN



Initialization:
We define the search range to be $[a, b] = [0, 60]$, with the end points 0 and 60 included in the search. We decide on the first search point by applying the golden section ratio 1/1.618:

$$b_1 = a + r(b - a) = 0 + 1/1.618(60 - 0) = 37.08.$$

Iteration 1:
Evaluation at the search point b_1 and another search point a_1 yields the following result: $g(a_1) > g(b_1)$, which means the optimal point has to be left of b_1, i.e., $x^* < b_1$. We discard the segment bb_1 from further consideration. The search point a_1 is defined as the proximal point to the left of b_1 generated from a golden section ratio distance from b, $a_1 = b - 0.618(b - a)$, where $0.618 = 1/1.618$. Notice the evaluation at the search point can be determined in a number of ways, including relatively subjective comparison between the two proximal search points, b_1 and a_1, regarding the preference between them, or a more formal market survey conducted for the two hypothetical locations. (See the airport location example in Section IV of Chapter 5.) But locating a_1 according to the Golden Section ratio would yield best results.

Our convention is to name the left point of the search interval a and the right point b. Here we switch our attention to the next poke point a_2 from our current position b_1: $b_1 \rightarrow a_2$, where the subscripts 1 and 2 denote the iteration number. Since our interval has been reduced from 60 to 37.08, we label b_1, the right end point of the search interval as b: $b = b_1 = 37.08$ and

$$a_2 = b - r(b - a) = 37.08 - 0.618(37.08 - 0) = 14.16.$$

Now we repeat this set of procedures iteratively.

Iteration 2:

$g(b_2) > g(a_2)$ means $x^* > a_2$, where b_2 is the proximal point to the right of a_2. The next poke point is $a_2 \rightarrow b_3$. The left end point is $a = a_2 = 14.16$, and $b_3 = a + r(b - a) = 14.16 + 0.618(37.08 - 14.16) = 28.33$.

Iteration 3:

$g(a_3) > g(b_3)$ means $x^* < b_3$, where a_Δ is the proximal point. The next poke point is $b_3 \rightarrow a_4$. The right end point is $b = b_3 = 28.33$, and $a_4 = b - r(b - a) = 28.33 - 0.6318(28.33 - 14.16) = \ldots$

Notice the interval of uncertainty regarding the location of the optimum retail location reduces steadily from 60: 60 → 37.08 → 23 → 14.16 → …

Stopping Rule:
When the interval of uncertainty gets down to a certain point, the algorithms stops. If we set the tolerance limit (or the error) to be ϵ, $b - a = 2\epsilon$ since the optimum is likely to be in the middle of the interval, everything else being equal. As an example: if ϵ is set at 0.04, stop when $b - a = 0.08$. When the interval of uncertainty gets down to this level, we terminate the algorithm. In this example, the retail shop location needs only be identified within 0.08 of a mile or 141 yards (127 m) along the highway.

E. Fibonacci Search Procedure

As mentioned previously, a more general procedure where the number of searches is limited is called Fibonacci search. Instead of positioning the search at a golden section ratio, we have a sequence of ratios for the first search point, the second search point, the third, and so on. Consider the following recursive relationship that generates an infinite series of numbers C_n which in turn determines such ratios:

$$C_n = C_{n-1} + C_{n-2} \qquad n = 2, 3, \ldots$$

Define $C_0 = 1$ and $C_1 = 1$. The above equation generates a series of numbers that are known as Fibonacci numbers:

Sequence k	*Identifier*	*Fibonacci No.* F_k
0	C_0	1
1	C_1	1
2	C_2	2
3	C_3	3
4	C_4	5
5	C_5	8
6	C_6	13
7	C_7	21
.	.	.
.	.	.

It can be shown that the Fibonacci search is an optimal search technique in the minimax sense. In other words, in a sequence of m functional evaluations, it will yield the minimum maximum interval of uncertainty.

Let Δ_k be the interval of uncertainty after k functional evaluations, and x_n be the decision variable x for which we seek an optimal value after k functional evaluations ($k = 1, 2, \ldots, m$). Unlike the Golden Section method, ϵ represents the given minimum separation allowed between any two points over the interval, instead of half the interval of uncertainty. In other words, ϵ represents the resolution that can be obtained experimentally between x_k and x_{k-1}. The initial interval is $\Delta_0 = b - a$. The evaluations at a and b, $f(a)$ and $f(b)$, yield no knowledge of where the optimal solution lies, preventing us from eliminating any region from the search interval. This means that the interval of uncertainty remains the same at the second iteration, or $\Delta_1 = b - a$. (In some ways, this is reflected through the first two Fibonacci numbers of 1.)

One can prove that the length of the interval of uncertainty after the first two functional evaluations is given by the following relationship:

$$\Delta_2 = \frac{1}{C_m}[\Delta_0 C_{m-1} + \epsilon(-1)^m] \tag{4.6}$$

The length of the final interval of uncertainty (which may not be less than ϵ) can also be shown to be given by the following equation:

$$\Delta_k = \frac{\Delta_0}{C_m} + \epsilon\frac{F_{k-2}}{C_m}$$

Notice the final interval of uncertainty is a function of the number of experimental evaluations (m), the allowable resolution (ϵ), and the initial search interval (Δ_0). The final interval will converge to zero as the number of functional evaluations increases to infinity, provided that ϵ is allowed to be infinitely small.

Finally, one can prove that the following evaluation is valid throughout the search procedure:

$$\Delta_k = \Delta_{k-2} - \Delta_{k-1} \qquad k = 3, 4, \ldots, m \tag{4.7}$$

Example

The Golden Section example was a good illustration, but it represents a rather symmetrical objective function $g(x)$ about the optimal location (x^*). Here we show another function which is a bit more skewed, just to demonstrate that the search technique can handle both situations, including the one illustrated in Figure 4.3(b). Let us get back to the example in which the accessibility measure is a function of mileage. We wish to maximize the calibrated accessibility function $f(x) = -3x^2 + 21.6x + 1.0$ between the interval [0, 25], with a minimum resolution of 0.50 and a search budget of six functional evaluations (Ravindran et al. 1987). In other words, the best accessibility is obtained somewhere between 0 and 2500 miles (4000 km) of additional highway built in the study area. Notice that such functions are not generally obtainable explicitly. It is given here for illustration purposes only.

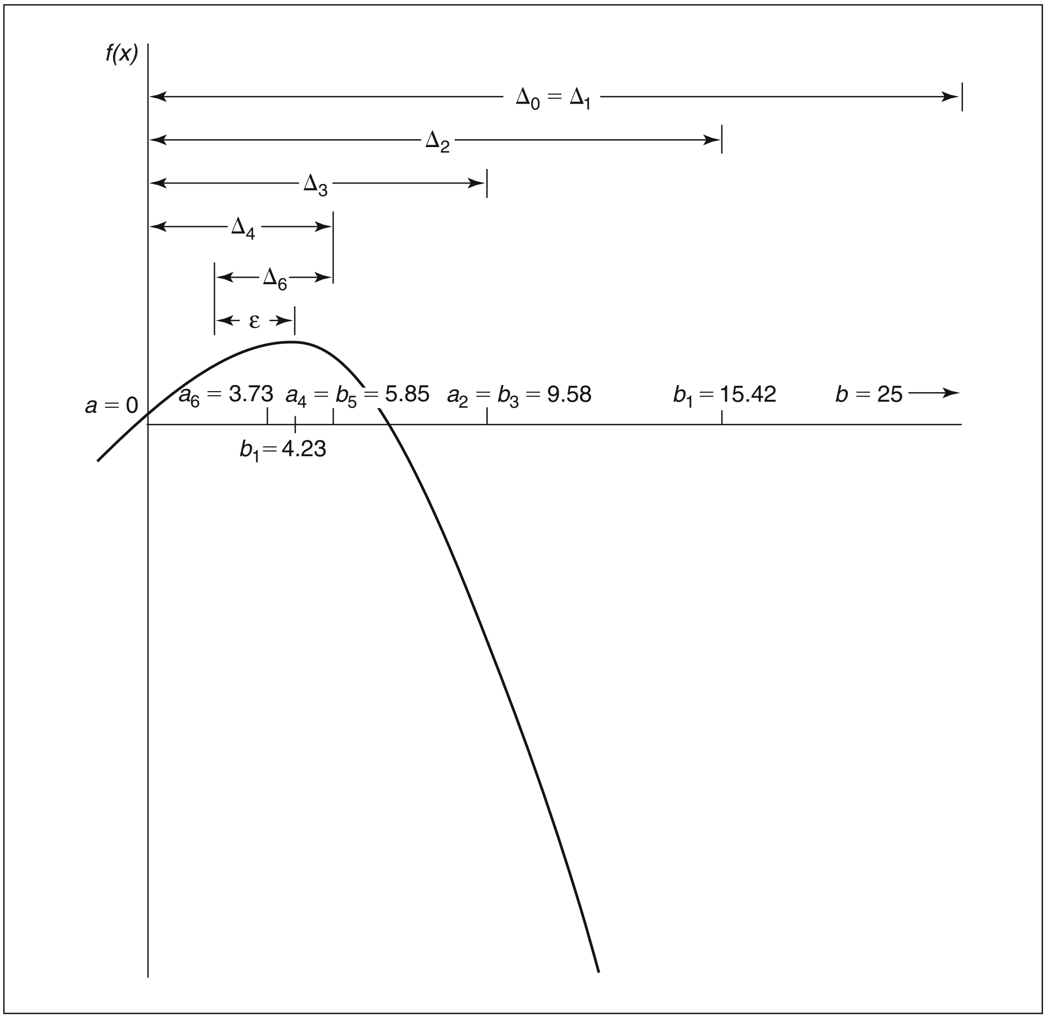
From equation 4.6,

$$\Delta_2 = \frac{1}{13}[25(8) + 0.50] = 15.4231$$

The first two functional evaluations will be conducted over the range [0, 25] symmetrical within this interval, where $b_1 = a + \Delta_2 = 0 + 15.4231 = 15.4231$ and $a_2 = b - \Delta_2 = 25 - 15.4231 = 9.5769$. This results in $f(b_1) = -\ 379.477$ and $f(a_2) = -\ 67.233$. Since the figure of merit is smaller at b_1 than at a_2, Figure 4.5 shows that the region to the right of $b_1 = 15.42$ can be eliminated. Note that $\Delta_0 = \Delta_1 = 25$. Hence $\Delta_3 = \Delta_2 - \Delta_2 = 25 - 15.4231 = 9.5769$ using Equation 4.7.

Symmetrical within the present interval of uncertainty, the two new points will be $b_3 = 9.5769$ and $a_4 = 5.8462$, and $f(b_3) = -67.233$, $f(a_4) = 24.744$. Notice that one of the new functional evaluations corresponds to one of the old functional evaluations. The current evaluation allows for the elimination of the region to the right of $b_3 = 9.5769$. The current interval of uncertainty is $\Delta_4 = \Delta_2 - \Delta_3 = 15.4231 - 9.5769 = 5.8462$. If we continue the process, convergence is obtained at the 6th iteration when the interval of uncertainty $\Delta_6 = 2.115$, and the resolution is $\epsilon = b_7 - a_6 = 4.2304 - 3.731 = 4994$, which is less than the specified minimum resolution of 0.5. In other words, the search now terminates

Figure 4.5 INTERVAL OF UNCERTAINTY IN FIBONACCI SEARCH



since an answer within 50 miles (80 km) is tolerable. Thus the best accessibility was obtained when 373 to 423 miles (597 − 677 km) of new highway are built. ■

From Equation 4.6

$$\underset{\substack{m \to \infty \\ \epsilon \to \infty}}{Lim}(\Delta_2) = \Delta_0[F_{m-1}/F_m]$$

One can show that in the limit the ratio F_{m-1}/F_m goes to 0.618, which is the golden section ratio 1/1.618. It is important to reemphasize once more that we do not need to know the precise form of the objective function in Fibonacci search. It is equally worthwhile to point out again that the evaluation of the figure of merit is oftentimes very difficult, even though we have assumed away the problem by having an analytical expression for the figure of merit *f*(*x*). This is where a search technique such as this comes in. In the two examples used, for instance, all that is necessary to evaluate the objective function is to compare it at two proximal points, and the only answer required is which is the better figure of merit, not by how much is it better. In the accessibility example, we simply perform traffic flow simulations at two proximal points to see which provides better area-wide accessibility. In the retail store location example, expert opinions can be used to compare the competitiveness of a store at one location vis-a-vis another, without explicitly quantifying the market share. Few optimization techniques would have this level of robustness and simplicity regarding the knowledge on the objective function.

III. ANALYTICAL SOLUTION TECHNIQUES

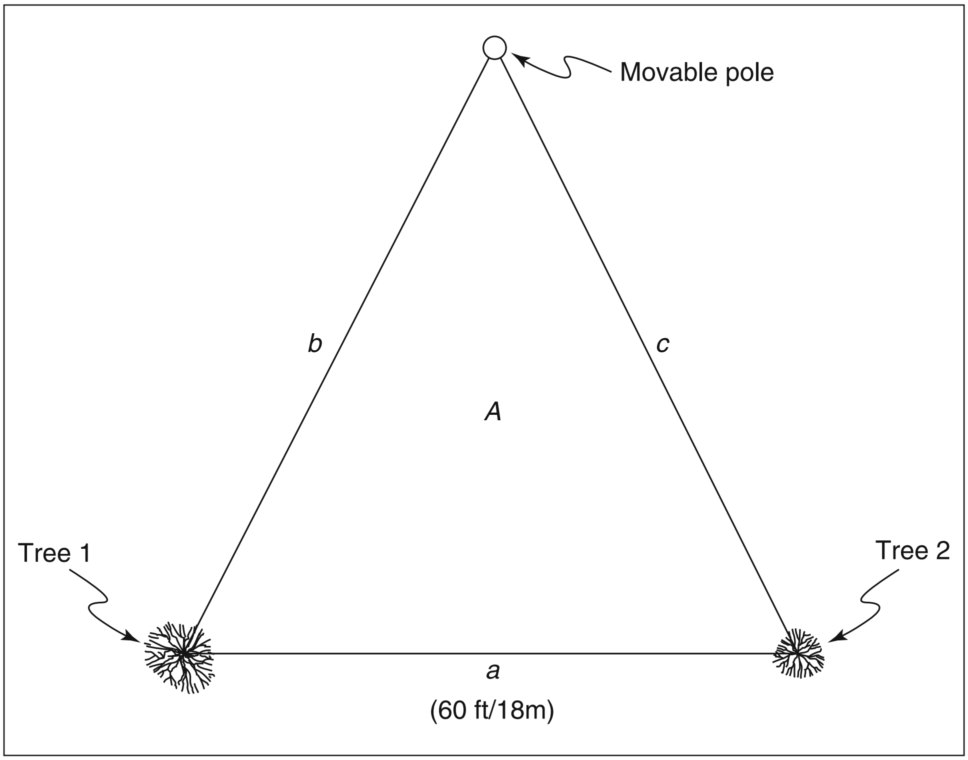
Analytical techniques, unlike the heuristic procedures, are subject to more rigorous mathematical treatment. They are usually solvable in closed form, rather than a process of trial and error (as was used in heuristic and direct search procedures). We have included here examples ranging from calculus to nonlinear programming.

A. Calculus

A familiar example of the analytical techniques is calculus. In this case, the objective function is expressible in a differentiable function, and the problem is subject to solutions by a well-defined set of theorems and procedures. It is required that there should be a peak or valley within the defined range of the variables. The reader might have already figured out that the accessibility maximization example above can easily be determined by setting the first derivative to zero: $\dot{f} \in (x) = 6x + 21.6 = 0$, or $x^* = 3.6$, which checks out with the previous solution using the Fibonacci search.

Instead of such an explicit functional form, an implicit function can be defined after the constraints are merged into the objective function, as is typically done in the **Lagrangian procedure.** Again the best way to illustrate this is through an example (Au and Stelson 1969). Suppose one is given a rope of *2s* feet (meters) and is to tie each end to a tree and to rope off an area as large as possible by locating a pole somewhere in the clearing, as illustrated in Figure 4.6. Where should the pole be placed? According to geometry, the area of a triangle *A* is defined in terms of its three sides—*a*, *b*, *c*—and the perimeter *2s* according to the

Figure 4.6 ROPING OFF AN AREA



following expression, which we maximize: Max $f(b, c) = A^2 = s(s-a)(s-b)(s-c)$. The only limitation is the length of the rope:

$$g(b, c) = a + b + c - 2s = 0 \tag{4.8}$$

The Lagrange procedure calls for the formation of a **Lagrangian function,** which is the linear combination of the objective and constraint, to be maximized:

$$\text{Max } L(b, c, \lambda) = f(b, c) - \lambda g(b, c) = s(s-a)(s-b)(s-c) - \lambda(a + b + c - 2s)$$

where λ is called the **Lagrange multiplier** or the **dual variable.**

Taking the first derivative of the Lagrangian function with respect to the three variables b, c and λ, we have

$$\dot{L}(b) = 0 \text{ yields } - s(s-a)(s-c) - \lambda = 0$$
$$\dot{L}(c) = 0 \text{ yields } - s(s-a)(s-b) - \lambda = 0$$
$$\dot{L}(\lambda) = 0 \text{ yields } a + b + c - 2s = 0.$$

Solving these three equations for three unknowns b, c, λ: $b = s-a/2$, $c = s-a/2$, and $\lambda = -as(s-a)/2$. Thus if the rope is 200 feet (60 m) in length, and the two trees are 60 ft (18 m) apart, an equilateral triangle should be formed as shown in Figure 4.6, where the two sides b and c each measures $(100) - (60/2) = 70$ ft (21 m). The area so enclosed is 1897 sq ft (171 m^2). As indicated by the Lagrange multiplier or (dual variable), $\lambda = \Delta z/\Delta c = \Delta z/\Delta b$. A movement of the pole to the left or the right of the existing position by one foot (0.3 m) of rope length will decrease the enclosed area by $[(60)(100)(100 - 60)/2]^{1/2} = 346$ sq ft (10.4 m^2).

In the section below and also in Appendix 4, we will discuss another analytical solution technique, linear programming. It will be shown that λ has the same interpretation as the dual variable in LP, meaning the effect of changing the resources on the right-hand sides (RHS) of the constraint equations (in this case 0) by $+\Delta$ or $-\Delta$ (where Δ is nonnegative in value). If $-\Delta$, the rope is lengthened according to Equation 4.8, and the area of the triangle will increase. On the other hand, if $+\Delta$, the rope is shortened, and the area will decrease instead. The interesting point is that $\lambda = \Delta{-z}/\Delta s$ is always nonnegative. The amount of increase or decrease is $\lambda\Delta$. Similar interpretation can be made for the distance between the trees a. In this case λ would be unrestricted in sign.

B. Linear Programming

There are other types of analytical techniques, including linear and nonlinear programming. First, let us discuss **linear programming** (LP), an optimization method involving a set of linear simultaneous equations. A classic LP model in the early development of urban modeling is the **Herbert-Stevens residential model** (Herbert and Stevens 1960). The model structures a market clearing mechanism (Devish et al. 2006) to allocate residential bundles among the wealthy and the poor (for example, $i = 1, 2$) over the zones k in an urban area (for example, $k = 1, 2$). We will use this model to illustrate the LP optimization technique. The first index in identifying variables and input parameters in the model is a household group i, distinguishing residents with different budgets and tastes ($i = 1, 2$ for the rich and poor as mentioned). Instead of recognizing specific families, certain groups (containing more than one family) with the same budget and tastes are considered. Next to be considered are certain types of amenities h associated with the housing, which may refer to the amount of green space or public services such as schools and hospitals ($h = 1, 2$). For a family type i, they are interested in the residential bundle with amenity level h in a specific zone k of the city, with an associated cost c_{ih}^{k}. Included in the aggregate cost c_{ih}^{k} are transportation expenditures considering the number of trips and the associated length of each trip, but excluding the site rent paid to the landlord.

Now let us discuss the land rent paid by group i for residence h. There are two types of rents, total site rent and unit site rent. Total site rent is the amount paid for the total housing while the unit site rent is just the amount of rent per unit acre (0.4 ha), where s_{ih}' stands for the acreage for housing type h considered by household type i. The former will be used in the primal version of the model while the latter will be used in the dual (the terms primal and dual will be explained below). Both of these costs are exclusive of the house and the travel cost; they refer to the land alone. Finally, the distinction between residential bundle and the market basket needs to be made. A residential bundle is simply an aggregate of the quality of the house, how nice the surroundings are, and the transportation cost. The market basket, on the other hand, is the residential bundle plus other commodities that can be fit into the residential budget, including land rent.

The LP model decides who is going to obtain a particular piece of land. In the allocation process, it recognizes that people try to economize and obtain the best for their residential budget b_{ih}, in other words, maximizing their savings $(b_{ih} - c_{ih}^{k})$. The savings can be applied toward paying site rent to the landlord, or it could be a net savings if land is free in an economic context (as in zones where a surplus of land is available). Thus we have made savings synonymous with rent paying ability—a household can afford to pay the landlord site rent only if it has savings. The objective function of the LP seeks to maximize aggregate rent paying ability over the entire study area. The final allocation is based on a tradeoff between how much one can afford and what one is looking for. These concepts can be formalized in the set of equations below.

Decision variables: $(\leftarrow x_{ih}^{k} \rightarrow)^T$ = transpose vector containing the number of households of group i using residential bundle h located in zone $k = (x_{11}^1 x_{12}^1 x_{21}^1 x_{22}^1)$ and $(x_{11}^2 x_{12}^2 x_{21}^2 x_{22}^2)$; it goes without saying that these variables in the two vectors should ideally be integers.

Input Coefficients: $(\leftarrow b_{ih} \rightarrow)^T$ = transpose vector of the residential budget (in 10,000 dollars) allocated by group i to bundle $h = (b_{11} b_{12} b_{21} b_{22}) = (5\ 4\ 3\ 2)$; $(\leftarrow c_{ih}{}^{k} \rightarrow)^T$ = transpose vector of annual cost (in 10,000 dollars) to group i who chooses bundle h in area k, exclusive of site cost (land rent) = $(c_{11}^1 c_{12}^1 c_{21}^1 c_{22}^1)$ and $(c_{11}^2 c_{12}^2 c_{21}^2 c_{22}^2)$ = (4 3 2 1) and (3 2 1 0.5) respectively; $(\leftarrow s'_{ih} \rightarrow)^T$ = transpose vector of the number of acres (ha) in the site used by a household of group i if it uses residential bundle $h = (s'_{11}\ s'_{12} s'_{21} s'_{22})$ = (0.9 0.8 0.6 0.5).

Right-Hand Sides: $(\leftarrow L^k \rightarrow)^T$ = transpose vector of acres (ha) of land available for residential use in zone $k = (L_1\ L_2) = (20\ 15)$; $(\leftarrow N_i \rightarrow)$ = transpose vector of the number of households of group i that are to be located in the study area = $(N_1\ N_2)$ = (15 10).

Now we can write out the set of constraint equations. For land availability, we have:

$$0.9x_{11}^1 + 0.8x_{12}^1 + 0.6x_{21}^1 + 0.5x_{22}^1 \leq 20 \qquad \text{in zone 1}$$
$$0.9x_{11}^2 + 0.8x_{12}^2 + 0.6x_{21}^2 + 0.5x_{22}^2 \leq 15 \qquad \text{in zone 2;}$$

Demand for housing can be written as:

$$x_{11}^1 + x_{12}^1 + x_{11}^2 + x_{12}^2 = 15 \qquad \text{for household group 1}$$
$$x_{21}^1 + x_{22}^1 + x_{21}^2 + x_{22}^2 = 10 \qquad \text{for household group 2.}$$

Finally we write the objective function:

$$\text{Max } (\textit{savings in rent}) = (5-4)x_{11}^1 + (4-3)x_{12}^1 + (3-2)x_{21}^1 + (2-1)x_{22}^1$$
$$+ (5-3)x_{11}^2 + (4-2)x_{12}^2 + (3-1)x_{21}^2 + (2-0.5)x_{22}^2$$

While this "toy example" is constructed for illustration purposes, its generalization to m residential bundles, n household groups and U' zones can be readily inferred. For the specific example above, the solution assigns 3.75 and

11.25 households in group 1 to residential bundle 1 in zone 1 and bundle 2 in zone 2 respectively, and 10 in group 2 to bundle 1 in zone 2. In other words, $x_{11}^{1} = 3.75$, $x_{12}^{2} = 11.25$, and $x_{21}^{2} = 10$ and all other decision variables are zero.[1] Thus housing type 1 is popular among residents in this area, and so is zone 2 as a place to live (Vernon et al. 1992). The latter appears reasonable since the cost coefficients for zone 2 are smaller than those in zone 1, resulting in the greatest increase in savings, defined, once again, as $(b_{ih} - c_{ih}^{k})$. Thus the model allocates as many households as possible in zone 2 and meets the remaining demand through the use of zone 1. An area-wide rent savings of $462,500 is achieved.

The above is called the **primal formulation** of the LP. The **dual formulation** is the mirror image of the primal and can be written after defining two dual variables written for the land-availability constraint and the demand-for-housing constraint respectively:

$$r^{k} = \text{rent per unit-of-land in zone } k \ (k = 1, 2)$$

$$v'_{i} = \text{subsidy per household in group } i \ (i = 1, 2)$$

Now the dual LP looks like:

$$\begin{aligned}
&0.9r^{1} - v_{1}' \geq (5 - 4)\\
&0.8r^{1} - v_{1}' \geq (4 - 3)\\
&0.6r^{1} - v_{2}' \geq (3 - 2)\\
&0.5r^{1} - v_{2}' \geq (2 - 1)\\
&0.9r^{2} - v_{1}' \geq (5 - 3)\\
&0.8r^{2} - v_{1}' \geq (4 - 2)\\
&0.6r^{2} - v_{2}' \geq (3 - 1)\\
&0.5r^{2} - v_{2}' \geq (2 - 0.5)\\
&\text{Min } 20r^{1} + 15r^{2} - 15v_{1}' - 10v_{2}'
\end{aligned}$$

While the rs are positive, the sign for the vs can be both positive or negative, since these dual variables are associated with the equality constraints defined for the demand for housing (as contrasted with the inequality constraints for land availability)—a point which will be elaborated shortly below.

The dual LP determines the rent in each zone k and the subsidy paid to each household group i. Let us explain more in detail. Landlords at each zone k can receive at least as much site rent per residential bundle h as the highest bidder of household group i is willing to pay. Please note the actual cost to a household is the rent a household i pays after accounting for the subsidy received by the household or a taxation on the household group (when v_{i}' is negative). The dual program minimizes total land rent paid to landlords in all the zones k, minus the subsidy to all household groups i—i.e., the net rent paid. Notice this may mean a certain amount of subsidy has to be paid to household group i in order to guarantee a location at a particular bid-rent. The availability of subsidy to household group i enables that household to locate a residential bundle h in neighborhood k—*a* location which would be impossible without the subsidy. A poorer household can in fact be the highest bidder per unit of land as long as the household bids on small lots. Likewise, subsidy may be assigned to a wealthy household to ensure a residential bundle location also. The use of subsidy variable v' sometimes presents a problem as one goes back and forth between the dual and the primal formulations. Take the primal formulation first: There may be situations where all of one household group i cannot be located in zone k due to the capacity constraint on the land L^k—under the primal objective function of maximizing total savings in location rents. The remaining households of group i have to be located elsewhere (in zone k' for instance). Relocating these group i households in k' zone, however, would involve a subsidy (viewing from the dual formulation). Because of the LP formulation, this subsidy v_i' must be assigned to *all* households in group i. This may lead to excessive high rents in the favorite zones, when the actual cost to a household is the net of actual rent minus the subsidy.

Solution to the dual LP yields $r^1 = 0$, $r^2 = 1.25$, $v_1' = -1.00$ and $v_2' = -1.25$. This says that surplus land is available in zone 1, resulting in rent per unit-of-land being zero in this zone.[2] The taxation for household group 1 is \$10,000 while that for household group 2 is \$12,500, being the wealthier of the two groups. It can be seen that the wealthy residents, similar to their less affluent counterparts, both want their desired housing type and location and are willing to pay for it. But the consumers' surplus, or the difference between the maximum amount that the consumer would pay and the amount the consumer actually pays, is distinctly different among the two. As expected in an LP, the dual solution of net rent paid over the study area is identical to the primal solution of total savings in land rent. Both are valuated at \$462,500.

C. Primal and Dual Linear Programs

The LP discussion highlights the most interesting relationship between the primal formulation of an optimization problem and its dual. For the same housing example, one can review the key features of this relationship by constructing the LP tableau contained in Table 4.3. In the tableau, it is clear that the dual formulation is simply the transposed primal tableau. The cost coefficients in the primal objective function become the right-hand side of the dual LP, and the right-hand side of the primal becomes the cost coefficients of the dual objective function. While we used to maximize in the primal, now we minimize in the dual. Each constraint of the primal has a dual variable assigned to it. Dual variables assigned to an inequality are positive in sign, while those assigned to equality constraints are unrestricted in sign as mentioned.

Table 4.3 PRIMAL AND DUAL TABLEAU EXAMPLE

Primal →	x^1_{11}	x^1_{12}	x^1_{21}	x^1_{22}	x^2_{11}	x^2_{12}	x^2_{21}	x^2_{22}	Min ↓
r^1	0.9	0.8	0.6	0.5					≤20
r^2					0.9	0.8	0.6	0.5	≤15
v_1'	−1	−1			−1	−1			=−15
v_2'			−1	−1			−1	−1	=−10
Max →	≥ (5 − 4)	≥ (4 − 3)	≥ (3 − 2)	≥ (2 − 1)	≥ (5 − 3)	≥ (4 − 2)	≥ (3 − 1)	≥ (2 − .5)	↑ Dual

Certain duality theorems govern the solutions of the primal and dual LPs:

(a) If both the primal and dual problems have feasible solutions, the primal problem has an optimal solution with a figure of merit equal to the dual problem.

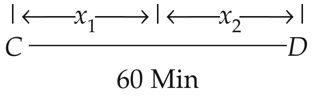
(b) Whenever a constraint in either one of the problems holds as a strict inequality so that there is slack (or surplus) in the constraint, the corresponding variable in the other problem equals zero. Otherwise a strict equality is obtained, together with the corresponding unrestricted variables in the other problem. This is usually referred to as primal and dual **complementary slackness.**

Based on statement (b) above, the dual variable, nonnegative in value, can be interpreted as the opportunity cost associated with the limited resource. In other words, r' corresponds to the additional contribution to the savings or rent figure of merit should land in zone 1 be increased by one unit. Thus both r^1 and r^2 can be interpreted as the additional net rent from an additional unit-of-land in zone 1. Parallel interpretation can be made regarding the dual variables associated with equality constraints, such as v_i'. In this case, the dual variable can be either positive, negative, or zero. In the Herbert-Stevens model above, for example, v_i' is negative and is interpreted as the taxation paid by household group i. Should v_i' be positive, it corresponds to subsidy, as mentioned previously. Both taxation and subsidy point toward the inherent valuation of group i toward their housing demand. Notice the dual variables are similar to the Lagrange multiplier in the discussion of calculus as an optimization technique. In fact, the Lagrange multipliers are the names given to dual variables in nonlinear differentiable functions through historical practice. The dual variable v_i' has a similar interpretation as the Lagrange multiplier λ in the example on roping off an area, as discussed under Section III-A. Both reflect the increase or decrease in objective function value should the strict equality constraint be relaxed.

D. Solution of Linear Programs

There are a number of ways to solve LP on the computer, ranging from traditional **simplex procedures** to newer techniques such as the **interior-point (projective)** method, from general procedures for regular tableaux to specialized techniques that exploit special structures of the tableau (See Appendix 4 or Bazaraa, Jarvis, and Sherali 1990). While the simplex algorithm is illustrated in Appendix 4, it is not the intent of this chapter to summarize all possible solution algorithms, nor are we in fact capable of doing so in such a limited space. Rather, we would like to highlight the salient points that will hopefully guide the location/land use analyst toward formulating a problem in an LP, selecting the appropriate computer package for the problem at hand, understanding the implications of the computer outputs, and perhaps most important of all, discerning abnormalities in the modeling process, if any, in a timely fashion. Let us use the same example we used in Chapter 1—the airport-location problem. Instead of the New York City area, let us move to the Midwest of the United States. Suppose an airport is to be built between Dayton, Ohio (population one million) and Cincinnati (population two million)—with a time separation of 60 minutes (Min) on Interstate Highway 75. We wish to locate the airport solely in such a way that the travel(measured in person-minutes) for all residents of the two cities is to be minimized. Where should we build the airport?

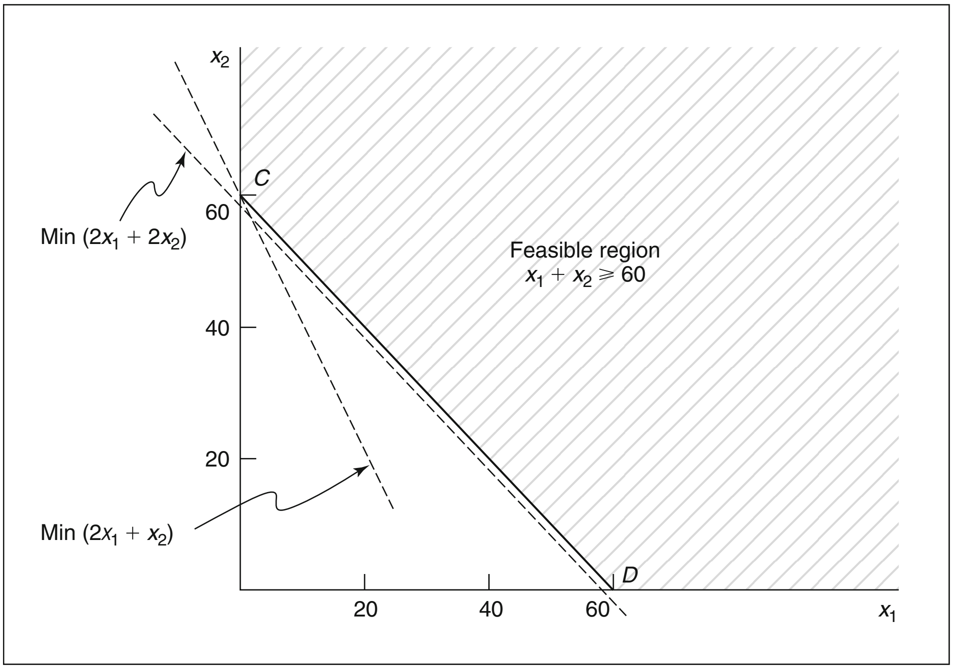
Let the airport be located x_1 Min away from Cincinnati (*C*) and x_2 Min from Dayton *(D).* The following LP can be constructed to model this problem: Min $\{2x_1 + x_2 \mid x_1 + x_2 \geq 60\}$, where the $\geq$ sign is used in the constraint to include the construction of an airport away from the Interstate Highway 75 that directly connects the two cities. The solution to this LP, in spite of its somewhat counter intuitive nature, is at either one of the extreme points *C* or *D,* in accordance with basic theorems in LP. This is shown below and in Figure 4.7, where the feasible region and objective function are plotted out in full. In this case, the airport is to be located at Cincinnati, $x^* = (0, 60)^T$, resulting in a minimum of 60 million total person-minutes of travel, $z^* = 60$.



Should the Dayton population grow to two million and the Cincinnati population remain at existing level, the LP now looks like: Min $\{2x_1 + 2x_2 \mid x_1 + x_2 \geq 60\}$. The multiple solution is shown in Figure 4.7. In this case, the airport can be anywhere between Dayton and Cincinnati on Interstate 75. Except for degeneracy, an LP solution algorithm amounts to an efficient way of implicitly (instead of exhaustively) evaluating the objective function at all possible extreme points and picking the very best. As mentioned, the simplex algorithm is described in Appendix 4 .

Such an analysis can be carried over to the case of three cities (Cincinnati, Columbus, and Dayton) and four cities (Cincinnati, Columbus, Dayton, and

Figure 4.7 GRAPHICAL SOLUTION OF AN AIRPORT LOCATION PROBLEM



Indianapolis). With Columbus' population at three million, Indianapolis at 3.5 million and the door-to-door times (after transportation improvement) as shown in Figure 4.8, these LPs were solved with the decision variables x_1, x_2, x_3, and x_4, corresponding to the time from Cincinnati, Columbus, Dayton, and Indianapolis respectively (Bartholomew, Brown, and Chan 1990; Cameron, O'Brien, and Chan 1990; McEachin, Taylor, and Chan 1992; Harry, Farmer, and Chan 1995).

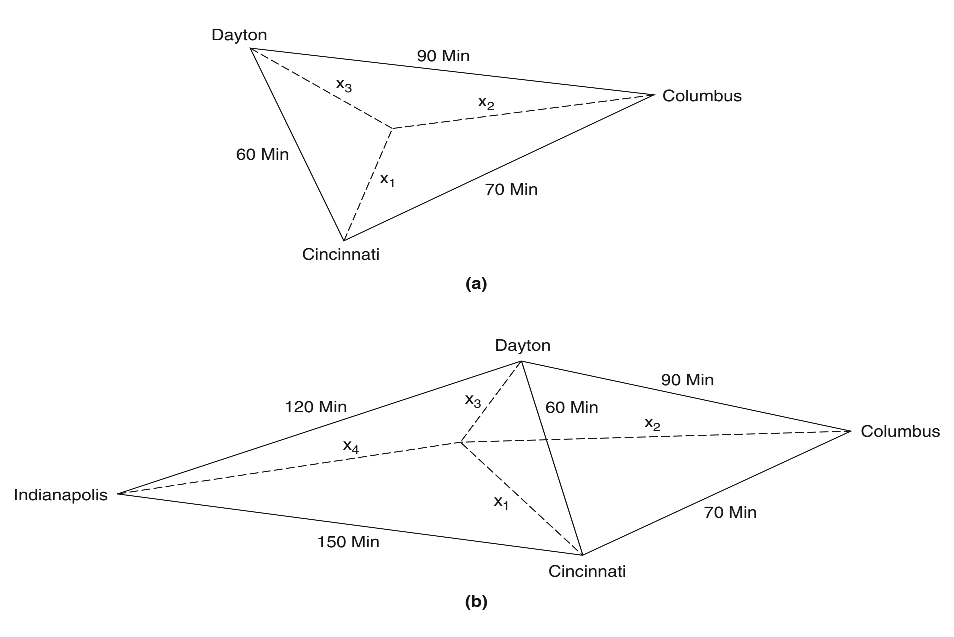
For the three-city case:

$$\begin{aligned} &\text{Min } (2x_1 + 3x_2 + x_3) \\ &\text{s.t. } x_1 + x_2 \geq 70 \\ &\quad\;\; x_1 + x_3 \geq 60 \\ &\quad\;\; x_2 + x_3 \geq 90 \\ &\quad\;\; x_1 + x_2 + x_3 \geq 125 \end{aligned} \tag{4.9}$$

where the last constraint shows minimum total time from each of the vertices of a triangle to a common point, as determined by the intersection of three angle bisectors (Claunch, Goehring, and Chan 1992). For the four-city case:

$$\begin{aligned} &\text{Min } (2x_1 + 3x_2 + x_3 + 3.5x_4) \\ &\text{s.t. } x_1 + x_2 \geq 70 \\ &\quad\;\; x_1 + x_3 \geq 60 \\ &\quad\;\; x_2 + x_3 \geq 90 \\ &\quad\;\; x_3 + x_4 \geq 120 \\ &\quad\;\; x_1 + x_4 \geq 150 \\ &\quad\;\; x_2 + x_4 \geq 206.62 \\ &\quad\;\; x_1 + x_2 + x_3 + x_4 \geq 266.62 \end{aligned} \tag{4.10}$$

Figure 4.8 THREE- AND FOUR-CITY EXTENSION OF THE AIRPORT LOCATION PROBLEM



where the last constraint represents the minimum total travel time from the four vertices to a common point, as determined by the bisectors from Cincinnati–Dayton, Columbus, and Indianapolis.

It was found that multiple solutions are again obtained in the three-city case, inasmuch as the combined Dayton and Cincinnati population exactly amounts to the Columbus population. Should one additional baby be born in Columbus, making Columbus the most populous city by just a shade, the airport would now be located in Columbus! In the case of four cities with the populations shown, the airport location changes to Cincinnati. Again the multiple solution is obtained when the population at all three or four of the cities are the same, where the multiple solutions occur at the inside of the convex hull formed by the cities as marked by the wedges.

Population (in millions) at					
Cincin	Columb	Dayton	Indianap	Airport location	Obj function value
2	3	1		$x_1 = 35, x_2 = 35, x_3 = 55$ or wedge *LD* in Figure 4.9 or $x_1 = 20, x_2 = 0, x_3 = 90$	*230* person-Min
2	3	1	3.5	$x_1 = 6.69, x_2 = 63.31,$ $x_3 = 53.31, x_4 = 143.31$ or wedge *LD* in Figure 4.10	*758.21* person-Min

In Figures 4.9 and 4.10, the figure legends suggest that each solution is qualified by a solution method and a model, each of which is denoted by a capital letter. Among the solution methods are LP, nonlinear program (NLP), direct search, and NLP version of the direct search. Among the models are the baseline

Figure 4.9 SOLUTIONS TO THE THREE-CITY CONFIGURATION

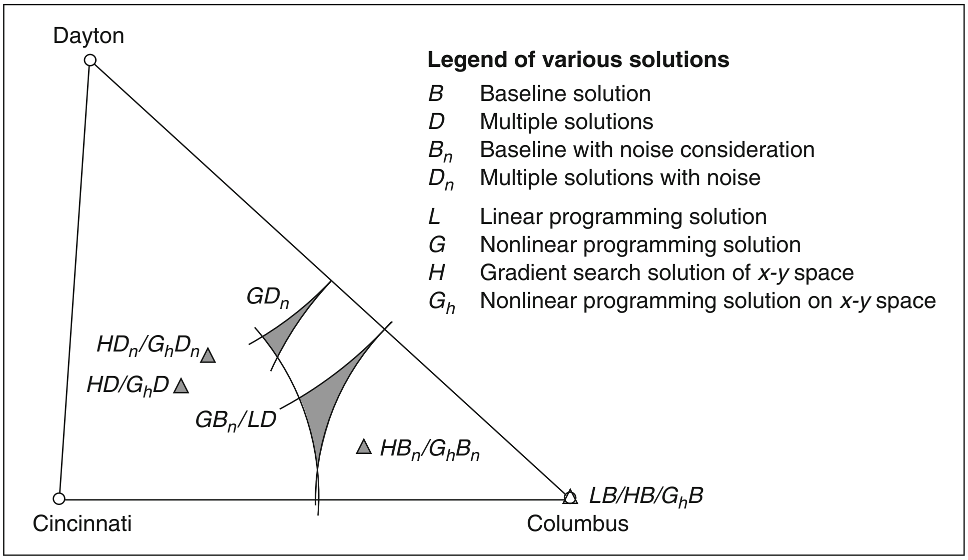
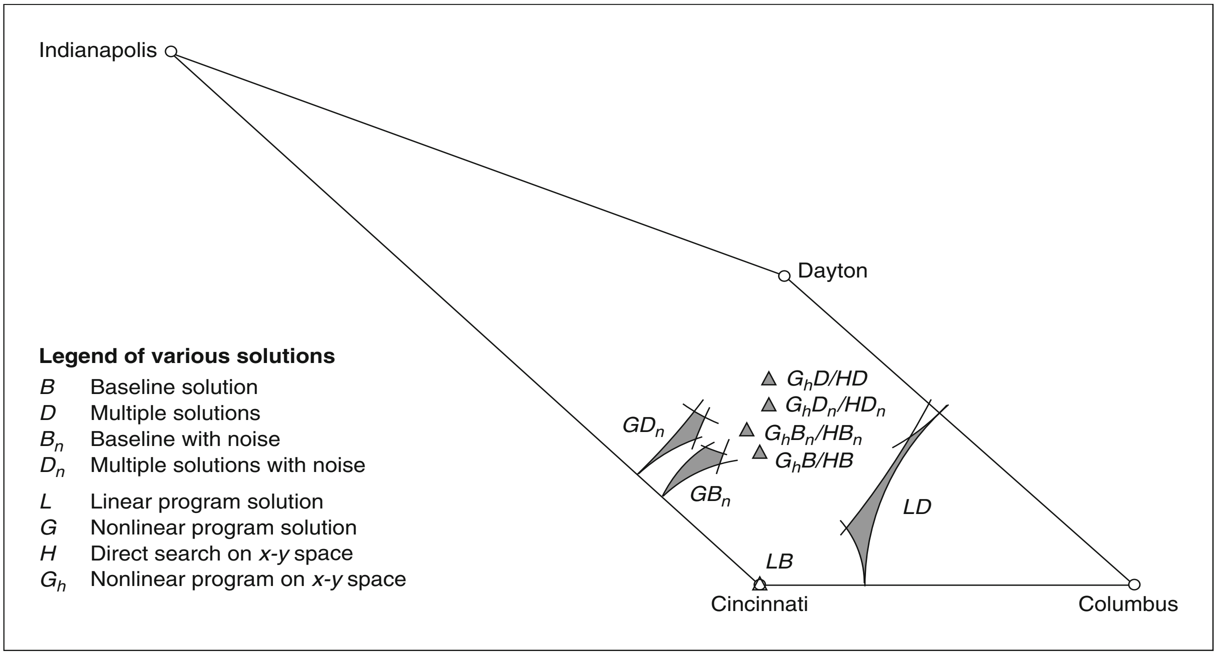

Dayton
Legend of various solutions
B Baseline solution
D Multiple solutions
Bn Baseline with noise consideration
Dn Multiple solutions with noise
L Linear programming solution
G Nonlinear programming solution
H Gradient search solution of x-y space
Gh Nonlinear programming solution on x-y space
GDn
HDn/GhDn
HD/GhD
GBn/LD
HBn/GhBn
LB/HB/GhB
Cincinnati
Columbus

Figure 4.10 SOLUTIONS TO THE FOUR-CITY CONFIGURATION


Indianapolis
Dayton
Legend of various solutions
B Baseline solution
D Multiple solutions
B_n Baseline with noise
D_n Multiple solutions with noise
L Linear program solution
G Nonlinear program solution
H Direct search on x-y space
G_h Nonlinear program on x-y space
G_hD/HD
G_hD_n/HD_n
G_hB_n/HB_n
G_hB/HB
GD_n
GB_n
LD
LB
Cincinnati
Columbus

solution with the given metropolitan population and both the multiple solutions when the combinations of populations are equal and when noise considerations are taken into account. Solutions to these LPs are very sensitive to the precise formulation and numerical errors, illustrating two key concerns in LP solution algorithms. Overall, these results are consistent with findings by Hurter and Martinich (1989), who reported studies in this Fermat or Steiner-Weber problem in the general context of industrial plant location.

Sensitivity analysis was performed by Leonard, McDaniel, and Nelson (1991), who changed the travel times between the cites slightly in the three-city problem. The travel times between Cincinnati–Columbus and Columbus–Dayton were reversed. The general result regarding extreme point and interior point solutions does not change. It appears that the driving force seems to be the populations rather than the travel times. Sensitivity analysis of the population coefficients of the three-city problem reinforces the observation that a city with a population larger than that of the remaining cities will host the airport. As mentioned, any increase in the Columbus population, which equals the combined populations of Cincinnati and Dayton, will make Columbus the optimal airport location. As another example, as soon as the Cincinnati population slightly dominates over Columbus and Dayton combined, the airport is located at 5 miles (8 km) outside the city. Sensitivity analysis on the four-city problem yields similar results. Finally, sensitivity experiments with the set of constraints yield interesting results. By deleting the last constraint of the three-city problem, a different and larger wedge of alternate solutions results. On the other hand, when the last constraint was removed in the four-city problem, the solution space did not change. Further examination shows that the last constraint is redundant and, hence, is not needed.

Before we conclude our discussion on linear programming, let us comment further on the computational aspects. Suffice to say that LP software has been perfected over the years. Numerical round-off errors involved in solving the linear set of equations has been an active area of investigation, resulting in steady improvements. Receiving equal attention is the storage requirement for intermediate data in large-scale problems, particularly regarding the basic feasible solutions corresponding to the various extreme points. Most recent advances have concentrated on input-output convenience (sometimes referred to as user friendliness). It is clear that the future generation of LP software (as well as nonlinear program software) will be those with symbolic-processing capabilities, including inputs expressed in algebraic forms such as equations and vector/matrix mathematical symbols (Brooke, Kendrick, and Meeraus 1995). In Chapter 6, we include an elementary example of such an input stream expressed in a set of equations. Equally viable is a parallel effort to link algorithmic procedures to data storage, as evidenced in spread sheet based procedures (Winston 1994).

E. Nonlinear Programming

When the objective function and/or the constraints are no longer linear functions, we have a **nonlinear program.** An example of a nonlinear objective function is the example used in Fibonacci search, where accessibility is given as a quadratic function of the highway mileage. Another example is the calculus optimization problem where the area enclosed by a rope is a nonlinear function of the rope length. These two examples can be considered special cases of nonlinear programming. Here we will examine the more general case and discuss a robust way of solving the general class of nonlinear programming problems. Again, we will use an example to introduce these concepts. Continuing the three-city airport study mentioned in the sensitivity analysis above, we introduce noise abatement as an additional concern (Leonard, McDaniel, and Nelson 1991). Here a simple representation of noise pollution is taken: *pollution* = (*constant*)(*population*)(*distance*)$^{-2}$. Following this assumption, the noise pollution at each city i is Kx_i^{-2}, where K is the calibration constant, same for all three cities. This term is added to the LP objectives of Equations 4.9 and 4.10, resulting in Equation 4.11. Different values of K were experimented with and at $K = 2{,}150$, the nonlinear objective function is equal to the linear at the halfway point between Dayton and Cincinnati in the two-city case. The constant K controls the effect of noise on the objective function. For very large K, the noise persists for a very long distance away from the airport. Conversely, as K approaches zero, the noise effect on the objective function becomes nonexistent. For values of K larger than 2150, two optimum locations were found, each located between the cities, symmetrically left and right of the center line (Interstate Highway 75). As K is reduced below 2,150, McEachin et al. (1992) found multiple optimal solutions. As a result of these experiments, the constant 2150 is used throughout the three- and four-city cases.

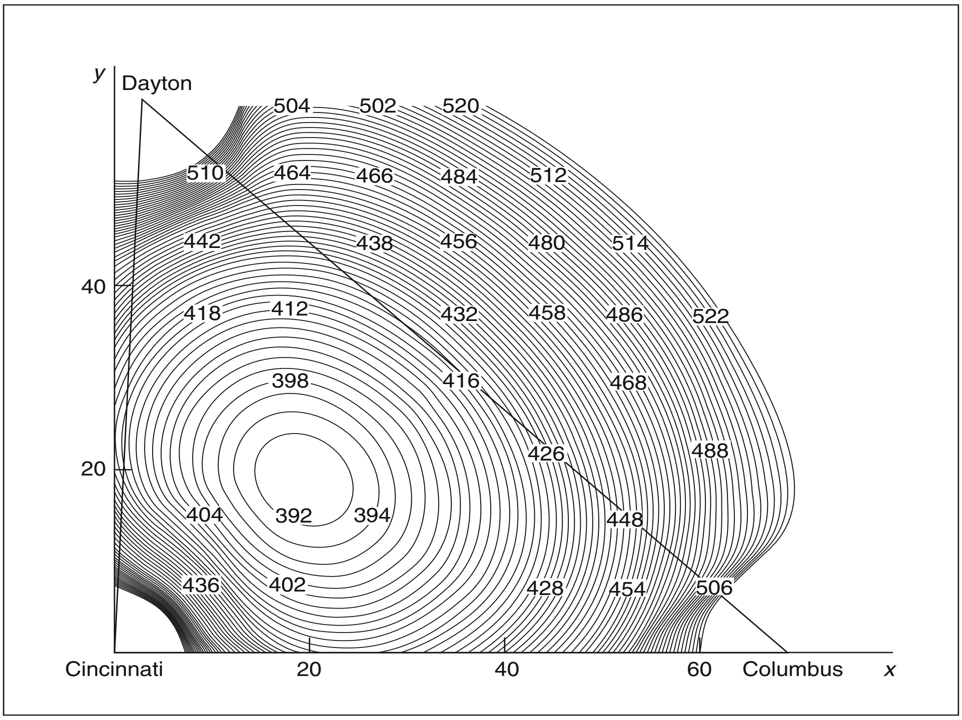
The three-city case now has the objective function

$$\text{Pollution} = 2(x_1 + Kx_1^{-2}) + 3(x_2 + Kx_2^{-2}) + (x_3 + Kx_3^{-2}) \tag{4.11}$$

Notice the objective function is a nonlinear function of the travel time decision variables. The airport location is now in the interior of the triangle defined by the three cities, at a point away from the three populations. When the populations at the three cities are equal, the airport location is at a point equally far away from each of the three cities. This interior point result is again consistent with Hurter and Martinich's general finding. Extensive computational results were obtained by McEachin et al. (1992) for the three- and four-city cases, as summarized in Figure 4.9 and Figure 4.10. Aside from an LP and NLP, a gradient search, or hill climbing algorithm was directly used to verify the results. A gradient search procedure is a general numerical way of solving nonlinear programs based on "climbing up the hill" in the steepest ascent direction in each step. The search was conducted directly on the triangle as defined in the *x-y* Euclidean space. The region within which the search is conducted is delineated by the three cities, or the four cities. An example of the gradient search solution for the three-city case is shown in Figure 4.11, complete with the combined noise and travel cost contours. Here one can see the optimum occurs at the "bottom of the valley" as defined by the contour 392 in this minimization example. Notice once again that the *x-y* space defines the feasible region for the search to be conducted, rather than the x_1, x_2 and x_3 decision-variable space used in both the LP and NLP models.

Similar to the linear case, the "legs" x_1, x_2, and x_3, as defined by the constraints shown in Equation 4.9 are not long enough to meet at a point in the three-city baseline configuration. The solution is somewhere within the triangular wedge as indicated by the symbol GB_n in Figure 4.9. The multiple solution where combinations of cities have the same population, is also shown as a similar wedge, labeled as GD_n. These solutions are obtained via two different solution methods. The first is an off-the-shelf, NLP solver GINO (Lasdon and Warren 1986). The

Figure 4.11 GRADIENT SEARCH SOLUTION TO THREE-CITY CONFIGURATION WITH NOISE



SOURCE: McEachin et al.(1992). Reprinted with permission.

second is the direct method of gradient search (Russell, Wang, and Berkhin 1992). The optimal NLP and direct search solutions of the baseline case diverge slightly. Their objective function values are close, at 239.47 and 248.12 respectively. The direct search solution is about six minutes away from the wedge defined by the NLP solution. Similar discrepancies are found between the multiple solutions. In our judgment, this reflects the numerical round-off errors by different algorithms, caused mainly by the difference between the feasible region defined by the x-y Euclidean space and the x_1, x_2, and x_3 space. To verify this point the analytical models were solved again using the NLP technique on the x-y space, rather than the x_1, x_2, and x_3 space. Both the direct search and NLP yield identical solutions. For completeness, we also include here the solutions for a four-city case in Figure 4.10, which exhibit many of the same phenomena.

F. Solution of a Nonlinear Program

We will illustrate two general types of solution algorithms. The first is a nonconvex programming solver based on the method of steepest ascent. This is intended for unconstrained optimization problems. Then we introduce the more general method to solve constrained problems.

1. Method of Steepest Ascent. As illustrated in Figure 4.12, the ideas behind the **method of steepest ascent** is quite simple. Starting with any initial point $\mathbf{x}^0$, one hikes up the mountain in the direction of steepest ascent. One keeps moving forward to the top of the ridge, at which time one reassesses the steepest ascent direction, which involves a 90-degree turn as shown at $\mathbf{x}^1$. Having re-established the steepest ascent direction, one again moves up to the top of the ridge at $\mathbf{x}^2$, takes another 90-degree turn and proceeds to move forward. If this procedure is repeated, one would eventually arrive at the top of the hill at $\mathbf{x}^*$. Notice we are on top of the hill instead of the mountain mainly because of the starting point $\boldsymbol{x}^0$. Should we start at $\mathbf{x}^{0\prime}$ instead of $\mathbf{x}^0$, one would have hiked up the top of the mountain at $\mathbf{x}^{**}$. We call $\mathbf{x}^*$ a local optimum and $\mathbf{x}^{**}$ a global optimum. Notice what we have performed is an unconstrained optimization. If one places a constraint such as $\boldsymbol{x}_1 = \boldsymbol{x}_2$ on this problem, the optimization result would have been different, the global maximum would have to be at $\mathbf{x}^*$ instead of $\mathbf{x}^{**}$ along the line $x_1 = x_2$.

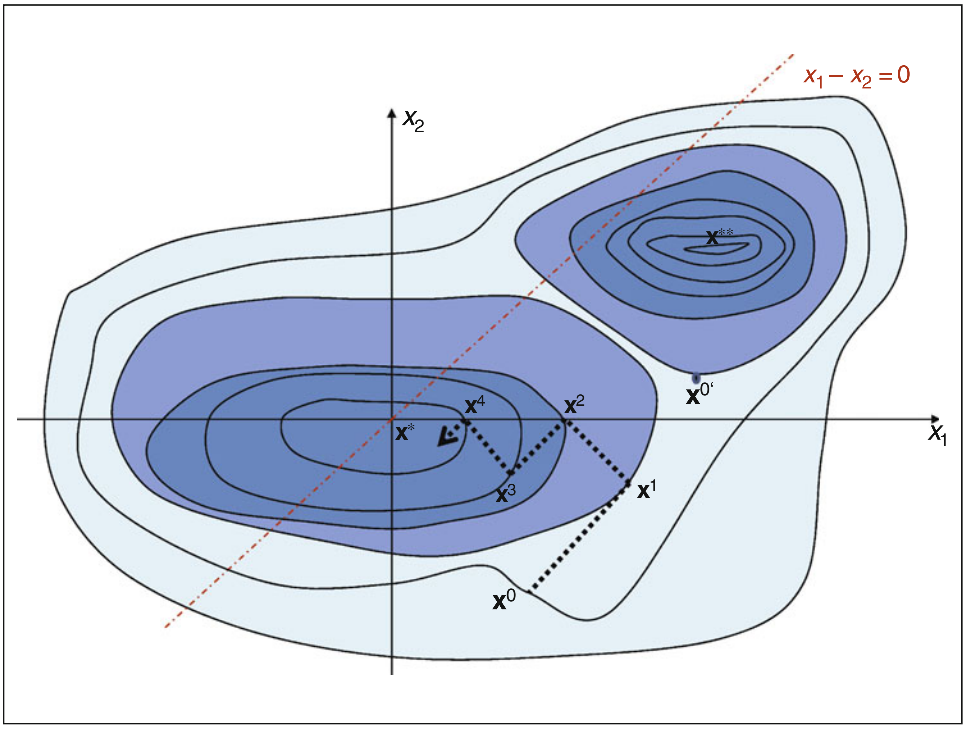
The general algorithm proceeds as follows:

1. Select a starting point $\mathbf{x}^k = \mathbf{x}^0 = (x_1^0, x_2^0, \ldots, x_n^0)$ and set $k = 0$.
2. Find a direction to move $\mathbf{d}^k = \nabla f(\mathbf{x}^k)$ which will improve (increase/decrease) the function at iteration k, where $\mathbf{d}^k = (d_1^k, d_2^k, \ldots, d_n^k)^T$
3. Move a distance t^k in the direction $\mathbf{d}^k$ to a new point $\mathbf{x}^{k+1} = \mathbf{x}^k + t^k\mathbf{d}^k$ where t^k is the nonnegative step size at iteration k, to be determined by (a) a line search (Golden Section for example), or (b) analytic technique (parametric in t^k).
4. Check for local optimality, for instance

$$\left.\frac{\partial f}{\partial x_j}\right|_{\mathbf{x}=\mathbf{x}^k} < \epsilon \qquad j = 1, 2, \ldots, n \tag{4.12}$$

If stopping criteria are not met, $k \rightarrow k + 1$, go to step 2, otherwise, stop.

Figure 4.12 EXAMPLE SEARCH



Note: Gradient is perpendicular to objective function contour at x^k and tangent at x^{k11} ($k = 0, 1, 2$).

Example

Suppose we wish to maximize the function $f(\boldsymbol{x}) = 2x_1x_2 + 2x_2 - x_1^2 - 2x_2^2$. The two components of the gradient are: $d_1 = \dot{f}(x_1) = 2x_2 - 2x_1$ and $d_2 = \dot{f}(x_2) = 2x_1 + 2 - 4x_2$, or $\boldsymbol{d}^0 = (d_1^0, d_2^0) = \nabla f(\boldsymbol{x}^0) = \nabla f(0, 0) = [\dot{f}(x_1 = 0, x_2 = 0), \dot{f}_{x2}(x_1) = 0, x_2 = 0)] = (0, 2)$. For $k = 1$, set $x_1^1 = 0 + t(0) = 0$, $x_2^1 = 0 + t(2) = 2t$. Then $f(\mathbf{x}^1) = t[\mathbf{x}^0 + t\nabla f(\mathbf{x}^0)] = f(0, 2t) = 2(0)(2t) + 2(2t) - (0)^2 - 2(2t)^2 = 4t - 8t^2$. Maximization of $f(x_1)$ over t yields $t^* = 1/4$, and correspondingly $\mathbf{x}_1 = (0,0) + 1/4(0, 2) = (0,1/2)$. Since $d_1 = 2(1/2) - 2(0) = 1$, it is clear that more iterations are necessary. Thus the iteration continues when we repeat what was applied toward $\mathbf{x}^0$ previously. ■

2. Karash-Kuhn-Tucker Conditions. Solution of a constrained nonlinear program is governed by the **Karash-Kuhn-Tucker (KKT) conditions,** which can be thought of as a generalization of the Lagrangian method discussed earlier. Consider the following optimization problem expressed in decision variable vector of n dimension in cartesian space:

$$\begin{array}{ll} \text{Max/Min } f(\mathbf{x}) & \\ g_i(x) = b_1' & i \in I' = \{1, 2, \ldots, m\} \end{array}$$

The Lagrangian method, as applied to equality constraints, can be represented in matrix algebra as

$$\nabla f(\mathbf{x}^*) -/+ \sum_{i \in I'} \lambda_i \nabla g_i(\mathbf{x}^*) = 0 \tag{4.13}$$

with the negative sign corresponding to the maximization problem and the positive sign minimization problem. This is sometimes referred to as the dual feasibility condition. Here ($\mathbf{x}^*$, $\boldsymbol{\lambda}$) is an optimal solution with unrestricted signs, in other words, both $\mathbf{x}$ and $\boldsymbol{\lambda}$ can assume either a positive or negative value. A complementary slackness relationship can also be written, similar to the relationship between a primal and dual LP:

$$\lambda_i [b'_1 - g_i(\mathbf{x}^*)] = 0 \qquad \text{for all } i \in I \tag{4.14}$$

These form the essence of the KKT conditions. Variants of these two equations can be written for nonnegative requirements on the variable **x** and **λ.**

Consider the mathematical program P'

$$\begin{aligned} &\text{Max/Min } f(\boldsymbol{x}) \\ &g_i(\boldsymbol{x}) \le b'_i \qquad i \in I' = \{1, 2, \ldots, m\} \end{aligned}$$

If $f(\boldsymbol{x})$, $g_i(\boldsymbol{x})$s are differentiable functions satisfying certain local regularity conditions such as non-singularity and convexity, then $\mathbf{x}^*$ can be an optimal solution to problem P' only if there exist nonnegative λ_i $(i \in I')$ such that the same conditions as the case with equality constraints apply. If $\mathbf{x}^*$ is to be nonnegative also,

$$\nabla f(\mathbf{x}^*) -/+ \sum_{i \in I'} \lambda_i \nabla g_i(\mathbf{x}^*) \le/\ge 0 \tag{4.15}$$

These necessary KKT conditions can be interpreted as a saddle point, as illustrated in Figure 4.13. Note Equations 4.13 and 4.14 hold for strict equality constraint irrespective of the sign of **x** and **λ** as in the Lagrangian, and according to Figure 4.13, partial derivatives with respect to x and λ are zero. On the other hand, for truncated xs and λs due to nonnegativity, we may fall short of the saddle point or the saddle point may not be reached.

The equivalent dual complementary slackness condition can be written in long hand as:

$$x_j \left[\frac{\partial f}{\partial x_j} - \sum_{i=1}^{m} \lambda_i \frac{\partial g_i}{\partial x_j} \right] = 0 \qquad j = 1, 2, \ldots, n \tag{4.16}$$

In the case of equality constraints, the dual feasibility condition (Equation 4.15), when expressed in long hand, becomes

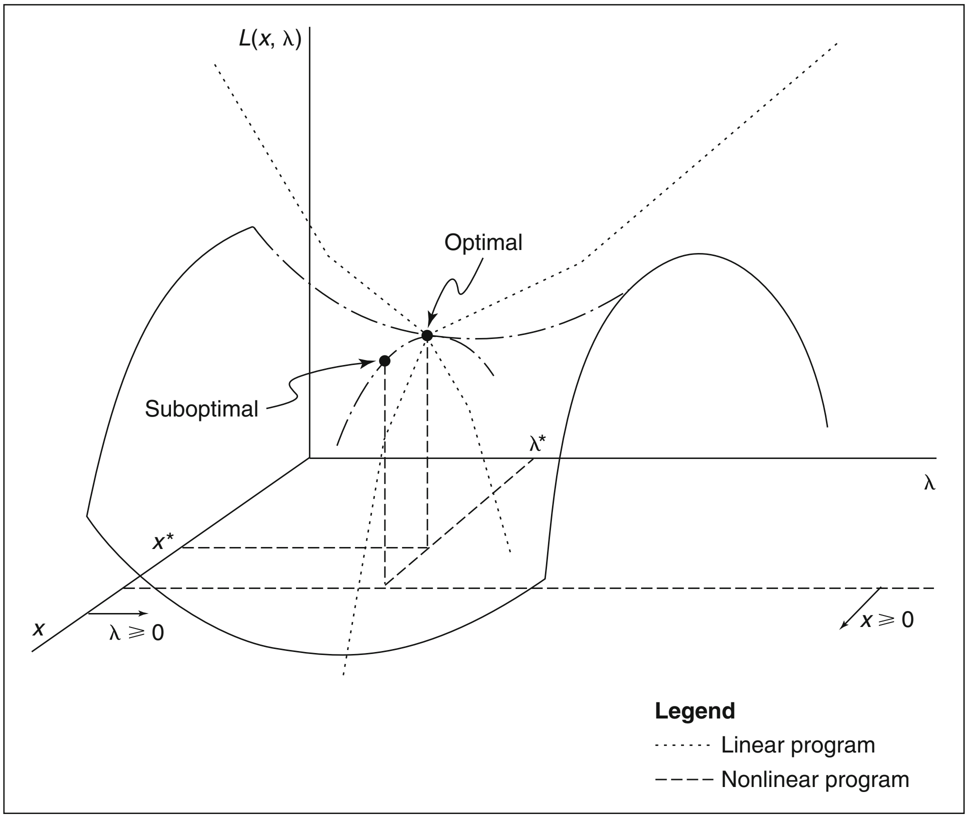
$$\frac{\partial f(\mathbf{x})}{\partial x_j} \le 0 \qquad j = 1, 2, \ldots, n \tag{4.17}$$

for maximization problems. The dual complementary slackness condition (Equation 4.16) above, in the case of equality constraints, would simply be

$$x_j \frac{\partial f(\boldsymbol{x})}{\partial x_j} = 0 \qquad j = 1, 2, \ldots, n \tag{4.18}$$

For illustration, one can check these KKT conditions against a two-variable LP.

Figure 4.13 MAX IN x AND MIN IN λ



$$
\begin{aligned}
&\text{Max } f(x) = c_1x_1 + c_2x_2 \\
&g_i(x) = a_{i1}x_1 + a_{i2}x_2 - b_i' \leq 0 \qquad (i = 1, 2) \\
&\mathbf{x}, \boldsymbol{\lambda} \geq \mathbf{0}.
\end{aligned}
$$

The first KKT equation (Equation 4.13) becomes the dual feasibility condition:

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \geq \lambda_1 \begin{bmatrix} a_{11} \\ a_{12} \end{bmatrix} + \lambda_2 \begin{bmatrix} a_{21} \\ a_{22} \end{bmatrix} = \overline{\mathbf{A}}^T\boldsymbol{\lambda} \qquad (4.19)$$

Pre-multiplying by $\mathbf{x}$, we obtain the weak duality $\mathbf{x}^Tc \geq \mathbf{x}^T\ \boldsymbol{\lambda}^T\boldsymbol{\lambda} = \mathbf{b}'^T\boldsymbol{\lambda}$ and the strong duality. In the former case, the primal objective function value is not less than the dual. In the latter case, the primal solution (z_x^*) is the same as dual solution (z_λ*) at optimality. The complementary slackness Equation 4.14 simply reads

$$
\begin{aligned}
\lambda_1\ (b_1' - a_{11}x_1 + a_{12}x_2) = 0 \\
\lambda_2\ (b_2' - a_{21}x_1 + a_{22}x_2) = 0
\end{aligned}
$$

3. Frank-Wolfe Method. Now that the KKT conditions have been briefly reviewed, let us go to NLP solution methods that build upon this primal dual relationship. Obviously, there are quite a few solution algorithms for an NLP. Some of them have been reviewed in the discussions on enumeration and calculus already.

A classic and rather robust method is that of **Frank-Wolfe** (F-W). For a continuously differentiable function $f(\boldsymbol{x})$, suppose we want to solve the NLP with linear constraints: Max/Min $\{f(\mathbf{x}) \mid \overline{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$. Here, $\mathbf{A}$ is an $m \times n$ tableau of coefficients, $\mathbf{x}$ is a vector of n decision variables and $\mathbf{b}'$ is the right-hand-side vector m long. The F-W algorithm linearizes the nonlinear objective function and turns it into an LP at each linearization. The algorithm converges to a solution after solving a series of LPs.

In detail, an iterative, primal method generates a sequence of points $\mathbf{x}^o, \ldots, \mathbf{x}^k \epsilon\ X = \{\mathbf{x} \mid \overline{A}\mathbf{x} = \mathbf{b}', \mathbf{x} \geq \mathbf{0}\}$ where $\mathbf{x}^{k+1}$ is found from $\mathbf{x}^k$ as follows: Set $k = 0$, start by solving the LP

$$\begin{array}{lll} \text{Max/Min } \nabla f^T(\mathbf{x}^k)\cdot\mathbf{x} & & \\ \text{s.t.} \qquad\quad \overline{A}\mathbf{x} = \mathbf{b}' & & \text{PL}(x^k) \\ \qquad\qquad\quad\ \mathbf{x} \geq \mathbf{0} & & \end{array}$$

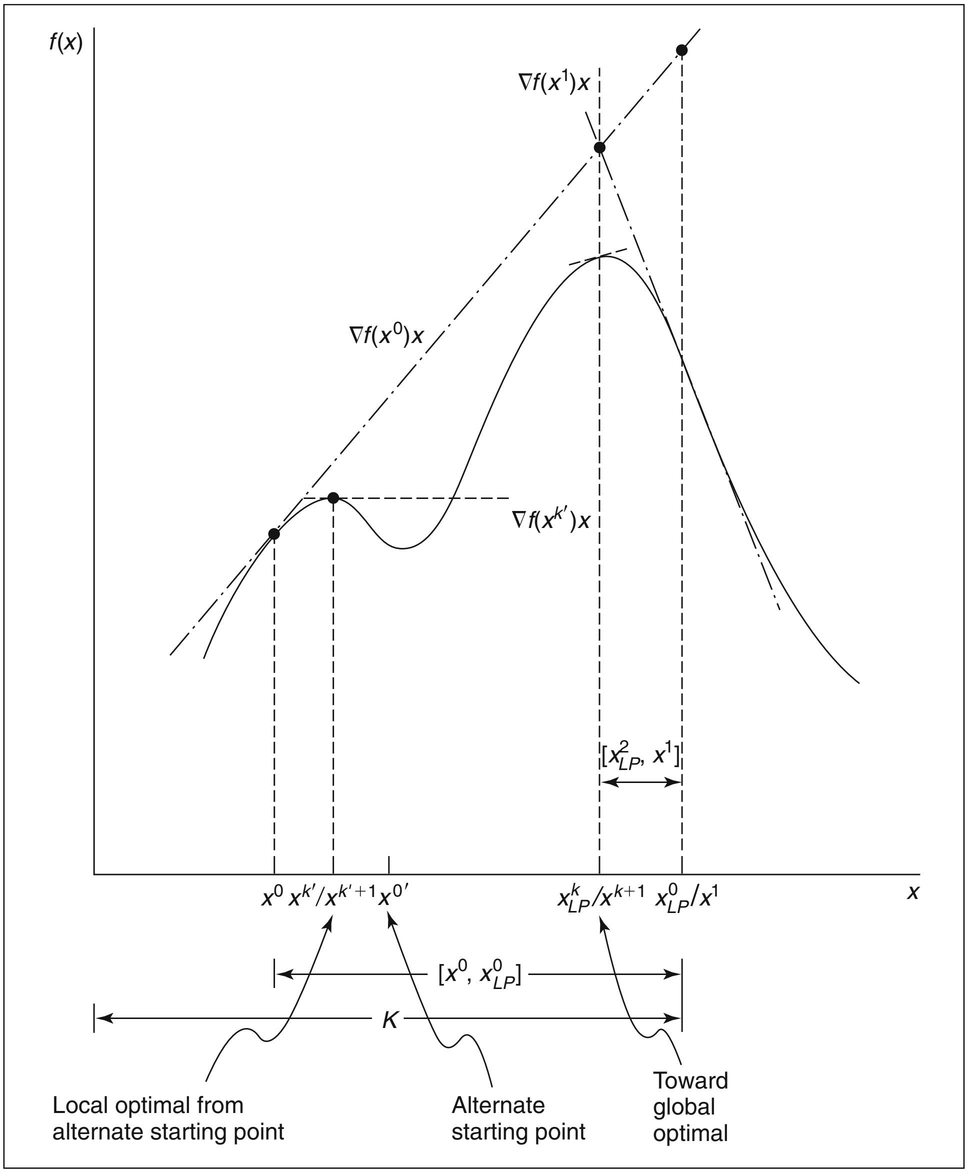
where $\nabla f^T(\mathbf{x}^k)$ is nothing more than the tangent at $\mathbf{x}^k$. It can be shown that the KKT dual complementary-slackness conditions—the mirror image of the primal ones outlined above in Equation 4.14—are included in the above LP: $[\nabla f^T(\mathbf{x}^o) -/+ \sum_{i \in I'} \lambda_i \nabla g_i(\mathbf{x}^o)]x^o = 0$ which, when $\mathbf{x}$ is non-zero, is equivalent to $\nabla f^T(\mathbf{x}^o)\mathbf{x}^o = \mathbf{b}'^T\boldsymbol{\lambda}$ and $\boldsymbol{c}^T\boldsymbol{x} = \boldsymbol{b}'^T\boldsymbol{\lambda}$ at optimality. Let $\mathbf{x}^k{}_{LP}$ be a vertex of X, an optimal solution of PL($\mathbf{x}^k$). Then $\mathbf{x}^{k+1}$ is chosen so as to maximize or minimize f in the interval $[\mathbf{x}^k, \mathbf{x}_{LP}]$. Figure 4.14 below will illustrate the algorithm, including the steps to convergence. It will be clear that the solution may be suboptimal—rather than globally optimal—depending on the starting point at which the algorithm is initiated.

In Figure 4.14, illustration of the generalized Frank-Wolfe algorithm is provided for a one-dimensional case, mainly for illustration clarity. We wish to maximize the accessibility function $f(x)$ over the interval of K miles (km) of additional highway. The starting point is x^0, where a tangent $\nabla f(x^0)x$ is constructed. It intersects the upper limit of our search interval K at x_{LP}^0—an extreme point of the LP. Our search interval now reduces from K to $[x^0, x_{LP}^0]$. Moving on to the first iteration of the algorithm, call this extremal point x^1. Again a tangent is constructed $\nabla f(x^1)x$, which intersects the tangent from the previous iteration $\nabla f(x^0)x$ at x_{LP}^1 ($k = 1$). Again, the interval reduces to $[x_{LP}^1, x^1]$. A tangent is constructed once more at this point, which we now rename x^2. The process simply continues until the peak at x_{LP}^k / x^{k+1} is reached. It is not hard to see that the peak so reached is a global maximum. On the other hand, an alternate starting point, say at $x^{0'}$, might end up at a local optimum such as $x^{k'} / x^{k'+1}$, with a tangent $\nabla f(x^{k'})x$ that is horizontal. Simple as it may be, this example illustrates how one can covert an NLP into a series of LP's and iteratively arrive at an optimum. Most important, this is a rather general procedure for a large number of situations, so long as $f(x)$ is differentiable. We will illustrate this procedure once more using the airport location problem in Chapter 5, where the one-dimensional example will be generalized.

IV. INTEGER OR MIXED-INTEGER PROGRAMMING

The next type of prescriptive technique to be discussed is **integer programming** (IP) or **mixed-integer programming** (MIP), where all integer solutions are required in IP while only some of the variables are required to be integer-valued in MIP. With

Figure 4.14 ONE-DIMENSIONAL ILLUSTRATION OF GENERALIZED FRANK-WOLFE ALGORITHM



this type of optimization, everything looks the same as an LP or NLP except a subset of the decision variables need to be discrete in the final answer. Examples of these have been given already in terms of housing location and industrial plant location. One recalls that in the Herbert-Stevens model, we ended up with integer value for only some of the households assigned to a particular housing type and a neighborhood. Should the answers be required in strict integers, it may be necessary to integerize them by a B&B procedure, where the branching rule is simply $\lceil x \rceil$ and $\lfloor x \rfloor$, corresponding to rounding a fractional variable to the next higher integer and next lower integer respectively. We will not discuss the details of the integerization outines inasmuch as they are all automated in IP or MIP codes. The logic is very much similar to the B&B procedure discussed previously in Section II-B when binary branches are constructed, wherein one branch fixes the fractional variable to the next higher integer and another branch the next lower integer as mentioned. A subproblem is defined as an LP corresponding to a relaxed version of the original IP or MIP, dropping its integrality requirements. Each of these subproblems is evaluated at each node of the B&B tree. In a nutshell, we solve the LP first and then fix the fractional variables, if they exist, by rounding the variable up or down. For this reason, the procedure is sometimes referred to as **LP relaxation,** to the extent that we relax the integrality requirement in the beginning. While this works reasonably well with small linear IPs or MIPs, the procedure is usually not advisable for large size problems, anywhere beyond several hundred integer variables at the time of writing. Neither does it work for NLPs where integer values are required, since the bounding rules become very nondiscriminating in such NLPs, thus resulting in a huge combinatorial space to be enumerated.

A. Total Unimodularity

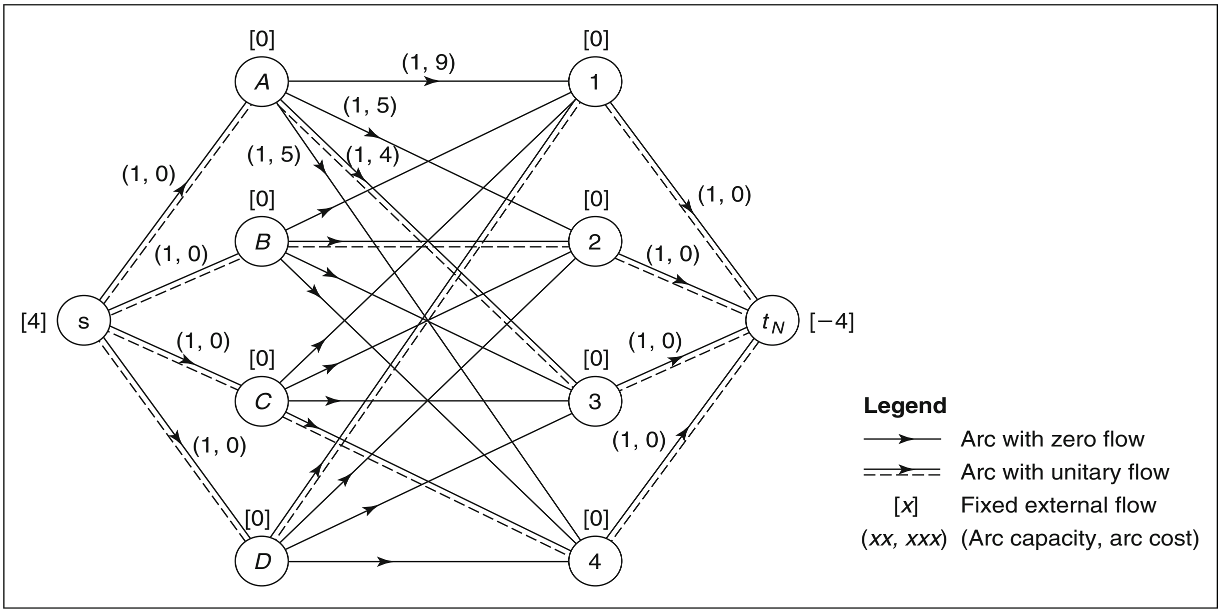
Let us now turn to the IP example corresponding to the location of industrial plants and look for ways other than B&B to generate integer solutions. We are required to have 0-1 variables x_{ij} in that example, corresponding to plant i being assigned to location j. The solution is guaranteed to be integer if one should solve the model simply as an LP since there are some inherent properties of this type of IP that are of interest: A matrix $\bar{\mathbf{A}}$ is said to be **totally unimodular (TU)** if and only if every subdeterminant of $\bar{\mathbf{A}}$ equals 1, − 1, or 0. A maximization (minimization) linear program with the constraint $\bar{\mathbf{A}}\mathbf{x} \leq (\geq)\mathbf{b}'$ and $\mathbf{x} \geq \mathbf{0}$ has an integer optimum solution for any arbitrary integer vector $\boldsymbol{b}'$ provided that the matrix $\bar{\mathbf{A}}$ is totally unimodular. It turns out that the tableau of the industrial plant location problem has the very exact TU property to make the solution integer, as one can see from the corresponding simplex tableau ($\bar{\mathbf{A}}$ matrix) for the industrial plant location problem in Table 4.4. This problem is related to the general class of transportation/allocation problems and the linear assignment problems.

In fact, we can characterize the problem as a network problem as well, since the problem can be represented as a bipartite network shown in Figure 4.15, with plants appearing in the left group of nodes and potential sites appearing on the right-hand side group (hence the term bipartite). A network tableau is by

Table 4.4 MATCHING TABLEAU FOR INDUSTRIAL PLANT LOCATION

Node	$(A, 1)$	$(A, 2)$	$(A, 3)$	$(A, 4)$	$(B, 1)$	$(B, 2)$	$(B, 3)$	$(B, 4)$	...	$(D, 4)$	RHS
A	1	1	1	1	1	1	1	1			1
B											1
C											1
D											1
1	−1				−1						−1
2		−1				−1					−1
3			−1				−1				−1
4				−1				−1		−1	−1

Figure 4.15 MATCHING NETWORK FOR INDUSTRIAL PLANT LOCATION



definition a TU matrix. Being able to cast a problem into a network model has additional computational advantages. Efficient codes exist for the solution of network models, particularly ones such as this, which are classified as pure Min-cost flow problems. Here one seeks the minimum cost flow from source s to sink t, directing four units of flow from left to right of the figure. Execution of these codes, including SAS/OR (1985, 1991) and CPLEX, will yield a solution such as the one shown in Figure 4.15. In the figure, unitary flows are found in the paths leading from s to t_n, valuating, among others, x_{A3}, x_{B2}, x_{C4}, and x_{D1} at unity. This assignment is identical to the one arrived at by B&B earlier in the chapter (in Section II-B).

Likewise, it can be shown that the Herbert-Stevens model can be cast into a network formulation. Raulerson, Bowyer, Zornick, and Chan (1994) have demonstrated that the problem can be structured as a bipartite graph as shown in Figure 4.15 by assigning the left group of nodes as the rich and poor resident groups and the right group of nodes as the housing types in each residential zone. The arc costs will simply be (c^1_{11}, c^1_{12}, c^2_{11}, c^2_{12}) emanating from the first left node and (c^1_{21}, c^1_{22}, c^2_{21}, c^2_{22}) emanating from the second node. In this case, there are two nodes on the left column and four on the right. In other words, the two columns no longer have the same number of nodes as in the plant location problem. A much faster execution time was observed in a sample problem for the Dayton, Ohio using the SAS/OR and CPLEX network software, rather than LP software.

B. Network Software

Many location problems can be formulated as flow and matching models, as alluded to above. These flow and matching models can in turn be solved efficiently when represented in terms of graphs and networks. We have already seen the efficacy of Min-cost-flow and assignment models in solving the industrial plant location problem above. By way of a definition, a **graph** is defined as a set of vertices or nodes V' connected by edges. A **network** is simply a graph with flow(s) on it. A directed graph has directionality placed on the edges, which are now referred to as arcs $\underline{A}$. When flow is placed on a directed graph (hence turning it into a network), the arcs are also called links. If a problem can be represented as a network, it can be solved readily by off-the-shelf network software, much like an LP can be solved by a variety of ready-made codes. Network problems also have physical analogues that one can relate to, such as the interpretation of dual variables v_i' as nodal potentials or odometer readings at node i (see Appendix 4).

Experiences with network software will show that the input is quite user-friendly. It follows a convention separate from regular LP. The arcs in the network, for example, are input in a head-to-tail format rather than the equivalent tableau format as shown in Table 4.4. An example for our industrial plant network in Figure 4.15 is given below.

From	*to*	*Cost*	*Capacity*
s	A	0	1
s	B	0	1
.	.	.	.
.	.	.	.
A	1	9	1
A	2	5	1
.	.	.	.
.	.	.	.
3	t_N	0	1
4	t_N	0	1

It can be shown that such a representation is equivalent to a specialized LP tableau **A** called node-arc incidence matrix. Such a matrix has been illustrated in Table 4.4 without the links from the source and without the links leading toward the sink. When represented in terms of a network, it is not surprising that a node-arc incidence matrix is TU. Most network codes use a set of terms that need to be explained. Several of these appear in Figure 4.15:

[x] = fixed external flow, a positive number means the injection
of flow into the network while a negative number depletion;
(xx, xxx) = (arc capacity, arc cost).

These network terms have close parallels to the equivalent LP. For example, fixed external flows correspond to the **RHS** vector of an LP tableau. In Table 4.4, for example, an equivalent representation of the network similar to Figure 4.15, without the arc emanating from source s and incident upon sink t_N, have fixed external flow of +1 and −1 at the individual sources A, B, C, D and sinks 1, 2, 3, 4 shown as the RHS. A master source and sink, while not necessary, are generally constructed for convenience—to make the problem solvable by generic, off-the-shelf network codes—as shown in Figure 4.15.

The arc costs in a network model correspond to the cost vector in the LP objective function. Thus a minimum cost flow will effectively represent the optimal solution to the equivalent minimization LP, with arc flow replacing the decision variables in the computation process. Arc capacity $\overline{u}_{ij}$, on the other hand, is the upper bound of a decision variable. Since the decision variables are 0-1 valued, it is not hard to see why a capacity of 1 is placed at each arc in the network model. Other equivalences can be established as well, but we will not have the space to go into them here. Interested readers may wish to consult Appendix 4 and excellent treatments of the subject in such texts as Ahuja et al. (1993) and Bertsekas (1991).

Aside from a network tableau, it is often not clear whether a large problem is TU, since the definition given above is far from an operational test. Experience has shown, however, that many sparse tableaux of mainly 0-1 entries often yield integer solutions as long as the right-hand side is integer. This is an important observation for the practitioners inasmuch as many facility location problems have exactly such a type of tableaux. Notice this means that a problem can be entered into a regular LP code with a good chance of obtaining an integer solution[3]. Modern codes such as CPLEX has the advanced feature to discern any network structure within a tableau and exploit it by employing network algorithms. The result is then combined with the non-network part of the tableau to provide the overall solution to the original problem. We have a detailed explanation of a network with side constraints algorithm in Appendix 4, where the side constraints refer to the non-network part of the tableau.

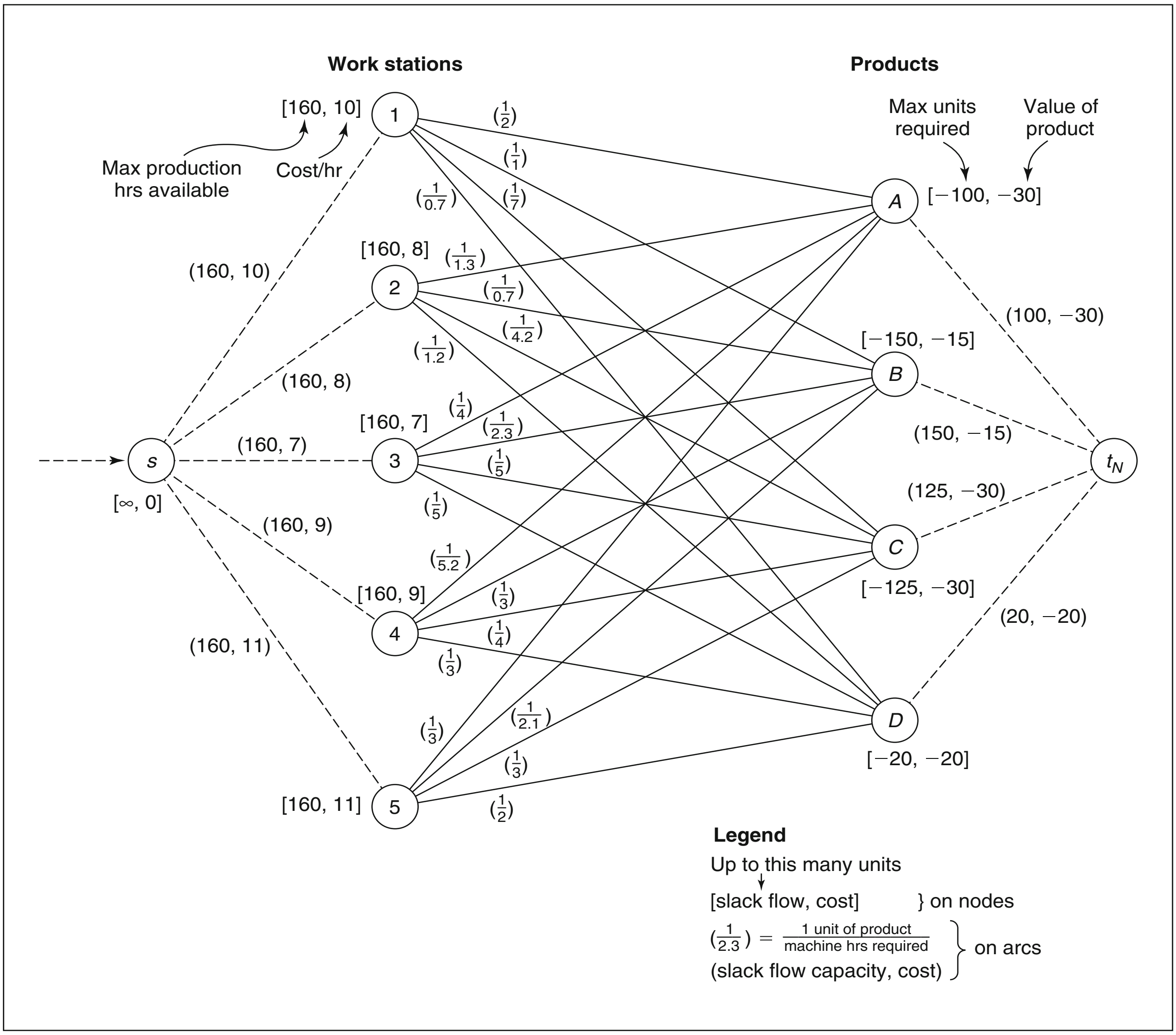
C. Network with Gains

Pure network flow models, functional as they may be, have only limited applications. Take the example of a material handling plant in which four products *A, B, C, D* are manufactured. These products can be made at any one of five work-stations 1, 2, 3, 4, and 5. Each work-station has a limited number of hours available for production and a specific unit cost of production. A minimum production quota is set for each product, a unit of which is valued at a certain amount. A network with gain flow model is used to solve the problem, with the intent of obtaining a most effective operation, in which the cost of manufacturing is minimized and the value of products is maximized. In Figure 4.16, the parameters at the supply nodes on the left column represent the maximum hours available at a work-station and the unit cost of production, while those at the demand nodes in the right column represent the maximum output potentials and the value of a product. The manufacturing plant operates for 20 eight-hour days each month (160 hours total) at the maximum. The arcs connecting the left column to the right column in this bipartite graph convert the flow in hours of production into units of product. This is accomplished by the gain parameter, which converts the hours of each work-station *i* to products at node *j*. The value of a product is expressed as a negative cost and the maximum number of production is expressed as a slack external flow.

This allocation model assigns work-stations to products so as to maximize value of production and minimize cost:

$$\begin{array}{lll} \text{Min} & (\Sigma_i\, w_i\, x_i + \Sigma_j\, w_j\, x_j) & \\ \text{s.t.} & \Sigma_j\, x_{ij} = x_i \leq 160 & \text{for all } i \\ & \Sigma_j\, a_{ij}\, x_{ij} = x_j \leq b_j & \text{for all } j \end{array}$$

Figure 4.16 NETWORK WITH GAIN MODEL OF PRODUCT ALLOCATION



where w_i is the cost/hr of work-station i, w_j is the value of product j (expressed in negative values), and $b_j' = 100, 150, 125, 20$ corresponding to production requirement for product type $j = A, B, C, D$. Notice $\Sigma_i\, x_i \neq \Sigma_j\, x_j$, hence the term network flow with the a_{ij} gain parameters (in this case it is actually loss). The total flow terminating at the sink t_N is less than the total flow emanating from the source s, in other words, conservation-of-flow no longer applies. The linear programming model above can be cast into a Min-cost-network-flow problem:

$$
\begin{aligned}
&\text{Min} \sum_{(i,\,j)\in \underline{A}} w_{ij}\, x_{ij} \\
&\text{s.t.} \sum_{j\in\delta^{+}(i)} x_{ij} - \sum_{j\in\delta^{-}(i)} a_{ji}\, c_{ji} = b_i' \qquad \text{for all } i \in V' \\
&\qquad 0 \le x_{ij} \le \overline{u}_{ij} \qquad\qquad\qquad\quad \text{for all } (i,\,j) \in \underline{A}
\end{aligned}
$$

where w replaces c as the symbol for costs, V' is the set of nodes and $\underline{A}$ a set of links, $\delta^+(i)$ is the set of nodes reachable from i, $\delta^-(i)$ is the set incident upon i, and $\overline{u}_{ij}$ is the capacity on arc (i,j). This is a more general network flow model than pure Min-cost-flow. For $a_{ji} = 1$, it reduces to the traditional pure Min-cost-flow model as discussed in the above section. On the other hand, when $a_{ji} \neq 1$, it represents a network with gain, as illustrated by the manufacturing example being discussed.

Let A be the matrix of constraint coefficients, the matrix formulation for this problem becomes Min$\{\mathbf{w}^T\mathbf{x} | A\mathbf{x} = \mathbf{b}', 0 \leq \mathbf{x} \leq \overline{\mathbf{u}}\}$. Notice that $\mathbf{A} = [\mathbf{A}_B, \mathbf{A}_N]$, where the node-arc incidence matrix written in terms of the basic and nonbasic parts, is no longer TU. As shown in Appendix 4, $\mathbf{A}_B$ is the basis matrix of $m - 1$ linearly independent columns, and $\mathbf{A}_N$ is the remaining nonbasic columns. $\mathbf{x}_B$ is a vector of basic flow variables in the same order as columns of $\mathbf{A}_B$, while $\mathbf{x}_N$ is the vector of nonbasic variables in the same order as columns of $\mathbf{A}_N$:

$$\mathbf{A}\mathbf{x} = [\mathbf{A}_B, \mathbf{A}_N]\begin{bmatrix}\mathbf{x}_B \\ \mathbf{x}_N\end{bmatrix} = \mathbf{b}'$$

$$\mathbf{A}_B\mathbf{x}_B + \mathbf{A}_N\mathbf{x}_N = \mathbf{b}'$$

and $\mathbf{x}_B = \mathbf{A}_B^{-1}[\mathbf{b}' - \mathbf{A}_N\mathbf{x}_N]$ through the **network simplex procedure,** which is explained in Appendix 4 in terms of the pure network flow version. In essence, the basis is represented via a tree consisting of $m - 1$ arcs. Instead of performing an algebraic basis inversion, we accomplish this through the tree graph. The fact that rank $\mathbf{A}$ = rank $(\mathbf{A}_B) = m - 1$ means $\mathbf{x}_B$ is unique.

Procedurally, the network with gain network algorithm will start with adding a node, called the slack node, and a number of additional arcs that enter or leave the slack node, called slack or **artificial arcs**. The slack node, usually numbered one greater than the number of nodes in the original network, can serve the dual function of a super source/sink. The flows associated with the slack node may not obey flow conservation (as amply illustrated by the current manufacturing example). The artificial arcs perform the function of the artificial variables of linear programming, and the slack arcs represent the slack external flows provided in the original model. As part of the network-simplex procedure, we already mentioned that the basis of the network flow LP is represented as the spanning tree of $m - 1$ nonzero arc flows, shown equivalently as $m - 1$ linearly independent columns of the node-arc incidence tableau. Here the basis of the LP is $m - 1$ in rank.

Let us now review the basic terms in a network with gains model. These definitions are best referenced against the manufacturing example illustrated in Figure 4.16: **External flow** enters or leaves a network, where **fixed external flow** b_i' at node i enters the network at (supply) node i if positive, and leaves the network (as a demand node) if negative. **Slack external flow**—a variable to be determined as part of the optimization procedure—is, once again, similar to slack variables in LP. A negative slack variable means extra supply available at node i to satisfy excess demands (as represented by the sum of flows incident upon i, $(\sum_{j\in\delta^-(i)} x_{ji})$, while a positive slack variable means extra demand available at node i to draw off excess supply $(\sum_{j\in\delta^+(i)} x_{ij})$. For a positive (negative) slack flow $x_{s''i}$ $(x_{is''})$ {or the external flow representations $[+x_{s''i}]$ $([-x_{is''}])$}, an arc is constructed from (to) the slack node s'' to (from) node i. The capacity of the arc is the absolute value of the slack flow capacity $|b_{s''i}|$, and the cost on the arc is $w_{s''i}$. The $b_{s''i}$s and $w_{s''i}$s are given (complete with positive and negative signs such as the network with gain example in Figure 4.16). If $b_{s''i}$ is positive (negative)

$x_{s''i}$ enters (leaves) the network as external flows. We will further illustrate the usefulness of this convention in Appendix 4. Case studies are documented in the "Facility Location" chapter of Chan (2005).

The **generalized network flow problem** is significantly more difficult to solve than the pure Min-cost-flow problem. The **generalized network simplex algorithm** is the fastest available algorithm for solving the generalized network flow problem in practice (Ahuja et al. 1993). The generalized network simplex algorithm is an adaptation of the LP simplex. This adaptation is possible because of the special topological structure of the basis. The basis of the generalized network flow problem is a **good augmented forest,** to be defined as follows: In the spanning tree with an extra arc called the root arc[4], an augmented forest is a collection of node disjoint augmented trees that span all the nodes of the graph. We define a **good augmented tree** as one whose cycle is formed by a gainy extra arc (in other words, an arc with a_{ij} associated with it) to the tree. A good augmented forest is consisted of nothing but good augmented trees. It can be shown that the basis of the generalized network flow problem is a good augmented forest. Good augmented forests play the same role in the generalized network simplex algorithm. Instead of one tree representing the basis, we have now more than one tree to span the nodes, depending on the number of gainy arcs involved. While the nodal potential equations are similar, the optimality conditions for a good augmented forest has a slightly different definition of the reduced cost, as defined by $w_{ij} - v_i + a_{ij}v_j$ (which is equivalent to $c_j - z_j$ in a regular simplex, with $c_j = w_{ij}$ and $z_j = v_j' - a_{ij}v_j$ in this case). Notice the equivalent pure-network-flow reduced cost is $w_{ij} - v_i' + v_j'$, where v_i', v_j' are the dual variables at rows i and j of the node-arc incidence matrix. v_i' can be interpreted as nodal potentials at node i, or alternatively as "odometer readings" in the context of measuring spatial separations. In the latter interpretation, a reduced cost of $w_{ij} - v_i' + v_j' \leq 0$ or $w_{ij} \leq v_i' - v_j'$ indicates a faster way to go from i to j via link (i,j), and hence the link should be used.

V. DECOMPOSITION METHODS IN FACILITY LOCATION

Consider this special MIP

$$\text{Min } (\boldsymbol{c}^T\boldsymbol{x} + \boldsymbol{g}^T\boldsymbol{y})$$

Subject to

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1, \ldots, m \tag{4.20}$$

$$x_{ij} \leq y_j \qquad i = 1, \ldots, m; \quad j = 1, \ldots, n \tag{4.21}$$

$$\sum_{j=1}^{n} y_j = p$$

$$x_{ij} \geq 0 \qquad i = 1, \ldots, m; \quad j = 1, \ldots, n \tag{4.22}$$

$$y_j = \{0, 1\} \qquad j = 1, \ldots, n$$

In this model, the continuous variable x_{ij} denotes the fraction of customer demand at node i that receives service from a facility at node j, while the 0-1 discrete variable y_j signifies whether or not a facility is built (Magnanti and Wong 1990). In network flow terminology, the forcing constraints Equation 4.21 restricts the flow to only those nodes j that have been chosen as facility sites. Finally, constraint Equation 4.22 restricts the number of facilities to a prescribed number p. This model is often referred to as the p-median problem, where c_{ij} denotes the cost of serving demand from node i by a facility at node j. It is conventional to set $\mathbf{g} = \mathbf{0}$, since oftentimes only the number of facilities, rather than their explicit facility costs, are of importance. The reader will recognize this as a special case of Benders' decomposition as discussed in Appendix 4, in which the decision variables can be decomposed into two groups: continuous and discrete. For reasons that will become clear, **Benders' decomposition** is also referred to as resource directive decomposition since it starts with a set of initial dual variables and adjust the common resource availability by fixing certain decision variables.

A. Resource Directive Decomposition

Consider the five-node, two-median example where the costs are defined as

$$[c_{ij}] = \begin{bmatrix} 0 & 5 & 9 & 11 & 12 \\ 5 & 0 & 4 & 6 & 7 \\ 9 & 4 & 0 & 2 & 3 \\ 11 & 6 & 2 & 0 & 1 \\ 12 & 7 & 3 & 1 & 0 \end{bmatrix}$$

where the entries specify the costs of serving the demand at node i from a facility located at node j. Suppose we have a current configuration with facilities located at nodes 2 and 5 (in other words, $y_2 = y_5 = 1$), the minimum cost objective function for this configuration is obtained by examining the minimum unit costs in columns 2 and 5. It is evaluated in this case at $(5 + 0) + (3 + 1 + 0) = 9$, suggesting that facility 2 serves demands at nodes 1 and 2 and facility 5 serves demands at 3, 4, and 5. Relative to the current solution, let us evaluate the reduction in the objective function cost if facility 1 is opened and all other facilities retain their current open-close status. This new facility would reduce the cost of servicing the demand at node 1 from $c_{12} = 5$ to $c_{11} = 0$. In other words, demand 1 will be served by facility 1 instead of facility 2, resulting in a cost reduction. Therefore the saving for opening facility 1 is 5 units. Similarly, by opening facility 3 we would reduce, relative to the current solution, cost of serving node 3 from $c_{35} = 3$ to $c_{33} = 0$. The saving is 3 units. Finally, opening facility 4 would reduce node 3 cost from 3 to 2 and node 4 cost from 1 to 0, for a total saving of $1 + 1 = 2$. Since facilities 2 and 5 are already open in the current solution, there is no saving for opening any of them.

Note that when these savings are combined, the individual assignments as computed above might overestimate possible total savings since the computation often double counts the cost reductions for any particular demand node. For example, our previous computations predict that opening both facilities 3 and 4 would reduce the cost of serving node 3 and give a total reduction of $3 + 1 = 4$ units, even though the maximum possible reduction is clearly 3 units, which is the cost of servicing node 3 in the current solution. (A demand can only be served by one facility, not both facilities.) With this savings information, we can bound the cost z of any feasible configuration y from below by

$$z \geq 9 - 5y_1 - 3y_3 - 2y_4 \tag{4.23}$$

Notice that specifying a different current configuration would change our savings computations and permit us to obtain a different lower bound function. For example, the readers can verify that configuration $y_1 = y_3 = 1$ would produce a lower bound inequality

$$z \geq 9 - 4y_2 - 4y_4 - 4y_5 \tag{4.24}$$

Each of these two bounding functions is always valid. By combining them we obtain an improved lower bound for the optimal two-median cost. Solving the following mixed integer program would determine the best location of the facilities that uses the combined lower bounding information:

$$\begin{array}{lll} \text{Min } z & & \\ \text{s.t.} & z \geq 9 - 5y_1 \qquad - 3y_3 - 2y_4 & \\ & z \geq 9 \qquad - 4y_2 \qquad - 4y_4 - 4y_5 & \\ & \quad y_1 + \ y_2 + \ y_3 + \ y_4 + \ y_5 = 2 & \\ & \quad y_j = \{0, 1\} \quad j = 1, \ldots, 5 & \end{array} \tag{4.25}$$

This yields a lower bound of $z' = 5$, obtained by setting $y_1' = y_2' = 1$ and $y_3' = y_4' = y_5' = 0$. Alternatively, one can set $y_1' = y_4' = 1$, or $y_1' = y_5' = 1$, or $y_3' = y_4' = 1$, and all other $y_j' = 0$ in each case. This bounding procedure is the essence of Benders' decomposition. In this context Equation 4.25 is referred to as a Benders' master problem and Equation 4.23 and Equation 4.24 are called Benders' cuts. When applied to an MIP with integer variables $\mathbf{y}$ and continuous variables $\mathbf{x}$, Benders' decomposition repeatedly solves a master problem like Equation 4.25 in the integer variables $\mathbf{y}$. At each step, the algorithm uses a simple savings computation to refine the lower bound information by adding a new Benders' cut to the master problem. Each solution $(z', \mathbf{y}')$ to the master problem yields a new lower bound z' and a new configuration $\mathbf{y}'$. For p-median problems, with the facility locations fixed at $\mathbf{y}'$, the resulting allocation problem becomes a trivial LP, *viz*, to assign all demand at node i to the closest open facility, or minimize c_{ij} over all j with $y_j' = 1$. It is in this resource allocation context that we refer to Benders' procedure as resource-directive decomposition. The optimal solution $\mathbf{x}'$ to this LP generates a new bound on the optimal objective function value of the p-median problem. As one will see in Appendix 4, the savings from any current configuration $\mathbf{y}'$ can be viewed as dual variables of this LP. Therefore, in general, the solution of an LP would replace the simple savings computation. The method terminates when the current lower bound z^* equals the cost of the best (least-cost) configuration $\mathbf{y}^*$ found so far. This equality implies that the best upper bound z equals the best lower bound z^* and so $\mathbf{y}^*$ must be an optimal configuration, with the associated optimal allocation $\mathbf{x}^*$. Again, the full Benders' decomposition algorithm is explained in Appendix 4.

B. Price Directive Decomposition

As contrasted with resource directive procedures, the dual variables of a price directive decomposition (or Lagrangian relaxation) are priced out in a Lagrangian, which decides on the next set of decision variables to be engaged. Lagrangian relaxation offers another type of decomposition technique that produces lower bounds. Consider the above *p*-median problem once again. As an algorithmic strategy for simplifying the problem, suppose we remove the constraints Equation 4.20, weighting them by Lagrange multipliers (dual variables) λ_i, and placing them in the objective function. We obtain the Lagrangian relaxation problem, namely

$$z_{LR}(\lambda) = \text{Min}\left[\sum_{i=1}^{5}\sum_{j=1}^{5} c_{ij}\,x_{ij} + \sum_{i=1}^{5}\lambda_i\left(1 - \sum_{j=1}^{5} x_{ij}\right)\right] \tag{4.26}$$

subject to the remaining constraints. Each penalty term $\lambda_i\left(1 - \sum_{j=1}^{5} x_{ij}\right)$ will be positive if λ_i has the appropriate sign and the *i*th constraint Equation 4.26 is violated. Therefore, by adjusting the penalty values λ_i, we can discourage Equation 4.26 from having an optimal solution that violates the constraints Equation 4.20.

Note that since the penalty term is always zero for all $\boldsymbol{\lambda}_i$ whenever $\mathbf{x}$ satisfies Equation 4.20, the optimal relaxation problem cost $z_{LR}(\boldsymbol{\lambda})$ is always a valid lower bound for the optimal *p*-median cost. The primary motivation for adopting this algorithmic strategy is that Equation 4.26 is very easy to solve. Let us set x_{ij} = 1 only when $y_j = 1$ or maintain feasibility of Equation 4.21. The modified cost coefficient $(c_{ij} - \lambda_i)$ of x_{ij} is nonpositive in Equation 4.26. Thus summing over all nodes *i*, the optimal benefit of setting $y_j = 1$ is $z_j = \Sigma_{i=1}^{5}$ [Min $(0, c_{ij} - \lambda_i)$] and we can rewrite Equation 4.26 as

$$z_{LR}(\boldsymbol{\lambda}) = \text{Min}\left[\sum_{j=1}^{5} z_j\,y_j + \sum_{i=1}^{5}\lambda_i\right] \tag{4.27}$$

subject to Equation 4.22, including the integrality and nonnegativity requirements. This problem is solved simply by finding the two smallest z_j values and setting the corresponding variables $y_j = 1$. For example, letting $\bar{\boldsymbol{\lambda}} = (3, 3, 3, 3, 3)^T$ for the time being, Equation 4.27 becomes

$$z_{LR}(\bar{\boldsymbol{\lambda}}) = \text{Min}\,(-3y_1 - 3y_2 - 4y_3 - 6y_4 - 5y_5 + 15) \tag{4.28}$$

The corresponding optimal solution for Equation 4.26 has a Lagrangian objective function value $z_{LR}(\bar{\boldsymbol{\lambda}}) = 15 - 6 - 5 = 4$; the solution has $y_4 = y_5 = 1$, $x_{34} = x_{44} = x_{45} = x_{54} = x_{55} = 1$, and all the other variables set to zero. Notice that this solution for the Lagrangian problem is not feasible for the *p*-median problem. Indeed, for any $i = 1, 2, \ldots, 5$, it does not satisfy the demand constraint Equation 4.20—hence the term relaxation in that certain constraints are ignored.

For another dual variable vector $\lambda^* = (5, 5, 3, 2, 3)^T$, Equation 4.27 becomes $z_{LR}(\bar{\boldsymbol{\lambda}}) = \text{Min}\,(-5x_1 - 5x_2 - 4x_3 - 5x_4 - 4x_5 - 18)$. Its optimal objective-function value $z_{LR}(\bar{\boldsymbol{\lambda}}) = 8$ is a tight lower bound since the optimal *p*-median cost is also eight. This example illustrates the importance of using good values for the dual variables λ_i in order to obtain strong lower bounds for the Lagrangian relaxation problem. In fact, to find the sharpest possible Lagrangian lower bound, we need to solve the optimization problem $\text{Max}_{\boldsymbol{\lambda}}\, z_{LR}(\bar{\boldsymbol{\lambda}})$. This optimization problem in the variables $\boldsymbol{\lambda}$ has become known as the Lagrangian dual problem to the original facility location model. (See Appendix 4 for a step-by-step algorithm to solve the Lagrangian relaxation problem.)

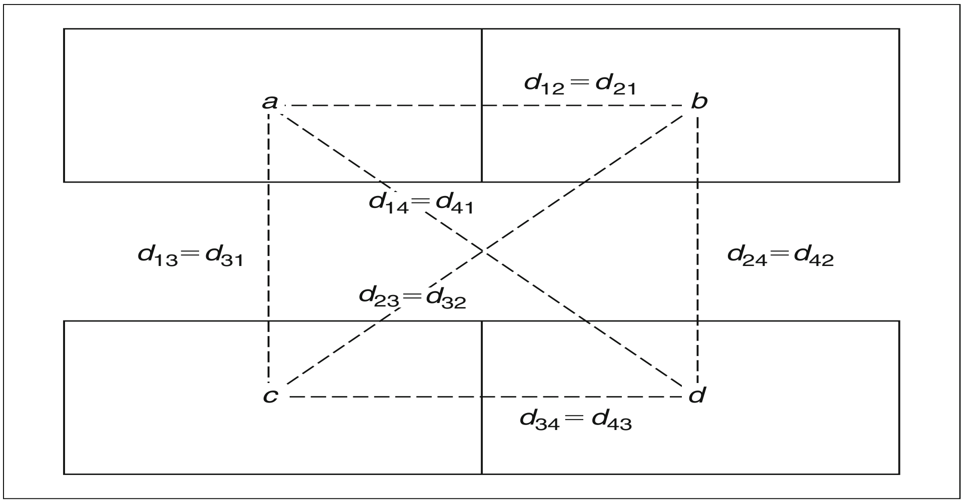
VI. SPATIAL INTERACTIONS: THE QUADRATIC ASSIGNMENT PROBLEM

In many locational problems the cost associated with placing a facility at a certain site depends not only on the distances from other facilities and the demands, but also on the interaction with other facilities (Burkard 1990; Francis, McGillis, Jr., and White 1992). In this section we examine a class of discrete location models that permit us to address certain interaction between facilities. The basic concepts can best be illustrated by an example: A manufacturing cell is being designed with manual material handling between work-stations. Figure 4.17 presents a schematic of the possible locations of the work-stations along the aisle. The objective is to minimize the total distance that material moves. The following distance matrix shows the separation between station locations *a, b, c, d* (in inches for example):

$$[d_{ij}] = \begin{bmatrix} 0 & 340 & 320 & 400 \\ 340 & 0 & 360 & 200 \\ 320 & 360 & 0 & 180 \\ 400 & 200 & 180 & 0 \end{bmatrix}$$

and material flow between the stations *A, B, C, D* themselves (in pounds for example) is represented by

Figure 4.17 LOCATION CONFIGURATION FOR WORK-STATIONS



$$[c_{kl}] = \begin{bmatrix} 0 & 80 & 40 & 30 \\ 80 & 0 & 30 & 20 \\ 40 & 30 & 0 & 10 \\ 30 & 20 & 10 & 0 \end{bmatrix}$$

What is the best assignment of work-stations to locations?

A. Nonlinear Formation

Define the discrete variable x_{ki} = 1 or 0, depending on whether work-station k is assigned to location i. Similarly, x_{lj} denotes whether work-station l is assigned to location j. Consider a piece of material moves from work-station k to work-station l, the total material movement (in, for instance, lb-ft or kg-m) per unit-time is to be minimized:

$$\text{Min} \sum_{k=1}^{3} \sum_{l=1}^{3} \sum_{i=1}^{3} \sum_{j=1}^{3} c_{kl}\, d_{ij}\, x_{ki}\, x_{lj}$$

Notice that the material flow between two locations i and j would depend on what work-stations are placed there. Since every work-station must be assigned to a location and every location must have a work-station, these constraints must be imposed:

$$\begin{aligned} &\sum_{i=1}^{3} x_{ki} = 1 && k = 1, 2, 3 \\ &\sum_{k=1}^{3} x_{ki} = 1 && i = 1, 2, 3 \\ &x_{ki} \in \{1, 0\} && k = 1, 2, 3; \quad i = 1, 2, 3 \end{aligned} \tag{4.29}$$

If we assume that station A is placed at location a, station B on location b, C on c, and D on d, then the objective function will assume the value

$$(2)[(340)(80) + (320)(40) + (400)(30) + (360)(30) + (200)(20) + (180)(10)] = 137{,}200$$

corresponding to the upper or lower triangle of the matrices $[d_{ij}]$ or $[c_{kl}]$. On the other hand, if we assume A is placed at d, B at b, C at c, and D at a, then we have

$$(2)[(340)(20) + (320)(10) + (400)(30) + (360)(30) + (200)(80) + (180)(40)] = 112{,}000$$

This comparison drives home the point that the cost of placing a work-station depends on the interaction with the other work-stations. In this case, the second configuration is better than the first since the amount of material movement is less.

B. Linear Formulation

This nonlinear program is computationally demanding. It has been shown that the model can be simplified to a linear integer program

$$\text{Min} \sum_{k=1}^{n} \sum_{l=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} r_{klij}\, y_{klij} \tag{4.30}$$

where $r_{klij} = c_{kl}d_{ij}$ and $y_{klij} = x_{ki}x_{lj}$. Notice the quadratic term $x_{ki}x_{lj}$ has now been replaced by a new binary variable y_{klij}. The constraints are the same as before except for two additional ones that govern the relationship between the x and y variables:

$$\begin{aligned} x_{ki} + x_{lj} - 2y_{klij} \geq 0 \qquad & k, l, i, j = 1, \ldots, n \\ y_{klij} \in \{0, 1\} \qquad & k, l, i, j = 1, \ldots, n \end{aligned} \tag{4.31}$$

Although the linear binary program was easy to implement, it has some worrisome properties. Along with the optimal solution for x_{ki}, we obtain the solution at the optimum for $y_{klij} = 0$. The objective function for the model, in turn, is zero. We are then required to take the optimal values of x_{ki}, $x_{14} = x_{25} = x_{32} = x_{41} = 1$ and substitute them into the original nonlinear objective function to obtain the final solution of 110,000 lb-ft (15,208 kg-m). This means that *A* is placed at *d*, *B* is at *c*, *C* on *b* and *D* on *a*.

For non-negative coefficients r_{klij}, the model can be recast into an MIP:

$$\text{Min} \sum_{k=1}^{n} \sum_{l=1}^{n} u_{kl},$$

where

$$u_{kl} = x_{ki} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} r_{klij}\, x_{lj} \right).$$

The constraints are the same as the original quadratic assignment problem, with these two additional ones that govern the relationship between x and u:

$$\begin{aligned} s_{kl}\, x_{ki} + \sum_{i=1}^{n} \sum_{i=1}^{n} r_{klij} x_{lj} - u_{kl} \leq s_{kl} \qquad & k, l = 1, \ldots, n \\ u_{kl} \geq 0 \qquad & k, l = 1, \ldots, n \end{aligned} \tag{4.32}$$

where

$$s_{kl} = \sum_{i=1}^{n} \sum_{j=1}^{n} r_{klij}.$$

Note that we can assume $r_{klij} \geq 0$ without loss of generality, since adding a constant to all cost coefficients does not change the optimal solution.

C. Comments

Many well-known combinatorial optimization problems can be formulated as quadratic assignment problems (QAP), including the linear assignment problem and traveling salesman problem (TSP). If a linear assignment problem with cost matrix $[c_{ij}]$ is given, let us define

$$r_{klij} \leftarrow \begin{cases} c_{ij} & \text{for } (i, j) = (k, l) \\ 0 & \text{otherwise} \end{cases} \tag{4.33}$$

or that the work-stations are permanently fixed at a unique location. Then the objective function of the given linear assignment problem is equivalent to the QAP objective (Equation 4.30). Because of space limitations presently, the readers are referred to Chan (2005) for the latter relationship between QAP and TSP. Suffice to say here that the TSP is defined as the least costly tour among *n* cities starting and ending in the same city, which on the surface bears little resemblance to the QAP, inasmuch as TSP is a routing problem and QAP is a location problem. The formal relationship between them and an extended explanation of TSP is offered in Chan (2005) under the chapters on "Routing" and "Location-and-Routing."

The QAP is discussed here for several reasons. It illustrates how a seemingly complex nonlinear formulation can be simplified by linearization. Indirectly, it also shows the importance of proper formulation to the solution algorithm, to the extent that a linear integer program or mixed integer program is much easier to solve than a nonlinear one. Thus while we laid out the various prescriptive techniques such as linear and nonlinear programming in this chapter, the distinction becomes blurred when we start solving problems in earnest. This does not mean the taxonomy proposed here is superficial, it simply argues for a deeper understanding of spatial temporal problems than the level exposed in general operations research textbooks. On the side, we suggested that different types of location and routing problems can be equivalenced also through transformation. Thus the linear assignment problem, the QAP and TSP all belong the same family. It will not be the first time we will see this type of discussion. In fact, a major focus of this book and Chan (2005) is to point out the mathematical equivalence between seemingly diverse physical spatial problems, including the land use model discussed in Section VIII.

VII. PRESCRIPTIVE ANALYSIS IN FACILITY LOCATION: DATA ENVELOPMENT ANALYSIS

Data Envelopment Analysis (DEA) is a linear programming technique to measure efficiency of a **decision-making unit** (DMU), interpreted here as alternative facility location (Thomas et al. 2002; Winston 1994). One of the major drawbacks of the traditional DEA formulation is that there is a separate formulation for each individual DMU. In other words, each site is separately modeled, and the results are then compared. Here we formulate a combined model that can assess the efficiency of several alternatives all at once. To complete the re-formulation of the traditional DEA model, additional variables must be introduced. One new variable x_i ($i = 1, 2, 3$) is the efficiency score of each facility. It carries a value between zero and one. A value close to one indicates an efficient DMU, or that a site is a viable alternative. The binary variable y_i ($i = 1, 2, 3$) indicates the selection of an alternative in the output. If an alternative *i* will not be included into the output, $y_i = 0$; otherwise it is unitary valued. Let b_{ki} denote the weights placed on the *k*th benefit of the *i*th alternative, and the weights c_{li} the *l*th cost for the *i*th alternative. The key is to note that b_{ki} and c_{li} will have a nonzero value only if $y_i = 1$. In the following facility-siting example, we will have three alternatives corresponding to three sites, three output benefit measures and two input cost measures. DEA simply picks the most efficient facility based on the benefit cost ratio.

The objective function now looks like

Maximize

$$x_1 + x_2 + x_3 + 0b_{11} + 0b_{21} + 0b_{31} + 0b_{12} + 0b_{22} + 0b_{32} + 0b_{13} + 0b_{23} + 0b_{33}$$
$$+0c_{11} + 0c_{21} + 0c_{12} + 0c_{22} + 0c_{13} + 0c_{23} + 0y_1 + 0y_2 + 0y_3$$

which seeks the highest, combined efficiency score among all three facilities, or $x_1 + x_2 + x_3$. The second step is to bound the efficiency scores by some large number M. In this example, M is set equal to 100. The number acts as a ceiling on each efficiency scores and prevents an unbounded answer.

$$x_1 \leq 100 \qquad x_2 \leq 100 \qquad x_3 \leq 100$$

This problem has only three solutions. Thus $y_1 = 0$, $y_2 = 0$, and $y_3 = 1$ is one of three solutions.

There needs to be some indication of the number of facilities to be included in the analysis. In this particular case, only one facility (the most efficient one) will be chosen:

$$y_1 + y_2 + y_3 = 1$$

The next set of constraints is used to compute the efficiency score of the facility in consideration. Here the benefits of each facility are evaluated. For example, facility 1 has 9 units of benefit 1, 4 units of benefit 2, and 16 units of benefit 3, and so on. Notice that the efficiency scores x_is are bounded above and below through the 0–1 ranged weights b_{kl}s and c_{li}s.

$$\begin{array}{ll}
x_1 - 9b_{11} - 4b_{21} - 16b_{31} \geq 0 & x_1 - 9b_{11} - 4b_{21} - 16b_{31} + 100y_1 \leq 100^{\#} \\
x_2 - 5b_{12} - 7b_{22} - 10b_{32} \geq 0 & x_2 - 5b_{12} - 7b_{22} - 10b_{32} + 100y_2 \leq 100 \\
x_3 - 4b_{13} - 9b_{23} - 13b_{33} \geq 0 & x_3 - 4b_{13} - 9b_{23} - 13b_{33} + 100y_3 \leq 100^{\&}
\end{array}$$

Inasmuch as the costs are bigger than benefits, the following set of constraints assures that the efficiency score for any efficient facility cannot exceed a value of 1. Notice that each facility has its own unique weighting system through the b_{ki}s and c_{li}s:

For facility 1

$$\begin{array}{l}
-9b_{11} - 4b_{21} - 16b_{31} + 5c_{11} + 14c_{21} \geq 0^{@} \\
-5b_1 - 7b_{21} - 10b_{31} + 8c_{11} + 15c_{21} \geq 0 \\
-4b_{11} - 9b_{21} - 13b_{31} + 7c_{11} + 12c_{21} \geq 0^{\%}
\end{array}$$

For facility 2

$$\begin{array}{l}
-9b_{12} - 4b_{22} - 16b_{32} + 5c_{12} + 14c_{22} \geq 0 \\
-5b_{12} - 7b_{22} - 10b_{32} + 8c_{12} + 15c_{22} \geq 0 \\
-4b_{12} - 9b_{22} - 13b_{32} + 7c_{12} + 12c_{22} \geq 0
\end{array}$$

For facility 3

$$\begin{aligned}
-9b_{13} - 4b_{23} - 16b_{33} + 5c_{13} + 14c_{23} &\geq 0\\
-5b_{13} - 7b_{23} - 10b_{33} + 8c_{13} + 15c_{23} &\geq 0\\
-4b_{13} - 9b_{23} - 13b_{33} + 7c_{13} + 12c_{23} &\geq 0
\end{aligned}$$

The next three constraints act as the scaling equations, which ensure that the weights are 0–1 ranged. The scaling is dependent upon whether or not the binary variable y_i is turned on or not. Thus if y_1 is unitary valued, the weights c_{11} and c_{21} are adjusted to convert the sum of the two costs associated with facility 1 (5 and 14) to be unity.

$$\begin{aligned}
5c_{11} + 14c_{21} - y_1 &= 0\\
8c_{12} + 15c_{22} - y_2 &= 0\\
7c_{13} + 12c_{23} - y_3 &= 0^*
\end{aligned}$$

Finally, the constraints which make this whole apparatus work are the weight forcing constraints. The constraints are either-or in nature and assure that the weights for a facility cannot be greater than zero unless the facility is turned on ($y_i = 1$). In addition, if the facility is turned on, it assures that each weight will be greater than a very small number. In the example, the number is assumed to be 0.0001.

Weight forcing constraints for facility 1

$$\begin{aligned}
b_{11} - 100y_1 &\leq 0 & -b_{11} + 100y_1 &\leq 99.9999\\
b_{21} - 100y_1 &\leq 0 & -b_{21} + 100y_1 &\leq 99.9999\\
b_{31} - 100y_1 &\leq 0 & -b_{31} + 100y_1 &\leq 99.9999\\
c_{11} - 100y_1 &\leq 0 & -c_{11} + 100y_1 &\leq 99.9999\\
c_{21} - 100y_1 &\leq 0 & -c_{21} + 100y_1 &\leq 99.9999
\end{aligned}$$

Weight forcing constraints for facility 2

$$\begin{aligned}
b_{12} - 100y_2 &\leq 0 & -b_{12} + 100y_2 &\leq 99.9999\\
b_{22} - 100y_2 &\leq 0 & -b_{22} + 100y_2 &\leq 99.9999\\
b_{32} - 100y_2 &\leq 0 & -b_{32} + 100y_2 &\leq 99.9999\\
c_{12} - 100y_2 &\leq 0 & -c_{12} + 100y_2 &\leq 99.9999\\
c_{22} - 100y_2 &\leq 0 & -c_{22} + 100y_2 &\leq 99.9999
\end{aligned}$$

Weight-forcing constraints for facility 3

$$\begin{aligned}
b_{13} - 100y_3 &\leq 0 & -b_{13} + 100y_3 &\leq 99.9999\\
b_{22} - 100y_3 &\leq 0 & -b_{23} + 100y_3 &\leq 99.9999\\
b_{33} - 100y_3 &\leq 0 & -b_{33} + 100y_3 &\leq 99.9999\\
c_{13} - 100y_3 &\leq 0 & -c_{13} + 100y_3 &\leq 99.9999\\
c_{23} - 100y_3 &\leq 0 & -c_{23} + 100y_3 &\leq 99.9999
\end{aligned}$$

For purpose of explanation, consider the weight forcing constraints for facility 1. The first five constraints in the set assure that the input and output weights will not be greater than zero unless $y_1 = 1$. If y_1 does indeed equal one, then the first five constraints also act as an acceptable upper bound on the expected values of the weights. Once again, an M value of 100 has been assumed. The second five constraints in the set for facility 1 assures that if the cost and benefit weights are turned on ($y_1 = 1$), they will be greater than or equal to 0.0001. The model output is very straightforward. Recall that the cap of 100 has been placed on all of the efficiency scores. Based on the number of facilities that are to be included in the analysis, the program will pick the facility or facilities that maximize the sum of the efficiency scores. For example, if only one facility is to be chosen, then only the most efficient facility will have its efficiency score calculated. The other efficiencies will be set to 100 (the upper bound), thereby indicating that the facility is not as efficient as the facility that was picked.

The linear programming solver confirmed the following results. The maximum objective function value is 201 (by definition). The problem has multiple optima. $y_1 = y_2 = 0$ and $y_3 = 1$ is one of the two optimal solutions. The weight-forcing constraints imply that $b_{11} = b_{21} = b_{31} = c_{11} = c_{21} = 0$, and $b_{12} = b_{22} = b_{32} = c_{12} = c_{22} = 0$. Efficiency-score-computation constraints and upper bounds on x_is imply that $x_1 = x_2 = 0$. The scaling equation (marked by *) implies that $7c_{13} + 12c_{23} = 1$. Constraints marked by # and & imply that $4b_{13} + 9b_{23} + 13b_{33} = x_3$. Hence the original problem with 55 constraints and 21 variables is reduced to the following problem:

$$\begin{array}{ll}
\text{Max} & 4b_{13} + 9b_{23} + 13b_{33} \\
\text{s.t.} & 5c_{13} + 14c_{23} \geq 9b_{13} + 4b_{23} + 16b_{33} \\
& 8c_{13} + 15c_{23} \geq 5b_{13} + 7b_{23} + 10b_{33} \\
& 7c_{13} + 12c_{23} \geq 4b_{13} + 9b_{23} + 13b_{33} \\
& 7c_{13} + 12c_{23} = 1
\end{array}$$

Now the question is: “Can we find weights for benefits and costs so that benefits are less-than-or-equal-to costs for all three facilities, and the benefits of facility 1 are equal to its costs (= 1)?” If the answer is affirmative, then the solution is called efficient. Here, the optimal objective-function-value is equal to 1, and thus the solution (locate at site 3) is efficient. In order to identify all efficient solutions, one must be able to identify all multiple optima to this LP. Notice DEA is a good illustration of a prescriptive model. The weights b and c are not determined through a consensus building process, but by prescription to push the efficiency envelope as far as possible. Further discussion of DEA in Facility Location is found in Chan (2005) under the “Spatial Separation” chapter.

Example

Solution of the above MIP suggests that facility 2 is inefficient and facilities 1 and 3 are efficient. Both x_1 and x_3 are unitary-valued. When one examines the optimal relaxed LP, the dual prices give us great insight into facility 2’s inefficiency. Consider all facilities whose efficiency constraints have non-zero dual-prices. (In our example, facilities 1 and 3 have non-zero dual prices.) If we form the weighted average of the output vectors and input vectors for these facilities (using the absolute value of the dual price from each facility as the weight), we obtain the following. Taking the dual variables from the weighting equations for each facility, namely Equations marked by @ and % above:

$$\begin{aligned}\text{Averaged output-vector} &= 0.26094\,(9\ 4\ 16)^T + 0.66003\,(4\ 9\ 13)^T\\ &= (4.98858\ 6.98403\ 12.75543)^T\end{aligned}$$

$$\text{Averaged input-vector} = 0.26094\,(5\ 14)^T + 0.66003\,(7\ 12)^T = (5.92491\ 11.57352)^T$$

Suppose we create a composite facility by combining 0.26094 of facility 1 with 0.66003 of facility 3. The averaged output-vector tells us that the composite facility produces close to the same amount of outputs 1 and 2 as facility 2 (5 and 7), but the composite facility produces $(12.75543 - 10) = 2.75543$ more of output 3. From the averaged input-vector for the composite facility, we find that the composite facility uses less of each input than does facility 2. We now see exactly where facility 2 is inefficient. ■

As explained in Chan (2005), DEA is a normative model to define a constant-return-to-scale *efficient frontier*, consisting of a set of *non-dominated solutions*—a term that is defined in the glossary of Technical Concepts (Appendix 5) as the "win–win" solutions. We arrive at these non-dominated solutions by combining the factor inputs in the correct proportions to achieve the best efficiency. Let us think of a set of isoquant lines as a measure of the efficiency deviation from an *ideal*, defined as a site that has the best of all attributes. DEA does not require the isoquant and its associated weights to be identified a priori. DEA determines the efficient frontier and optimal weighting scheme during the execution of the linear program. When DEA inputs and outputs are interpreted as costs and benefits, it can be used for finding the "best" facility location. In our presentations so far in this book, location and DEA models have been solved separately. A closer examination of the above DEA model used to site facilities also reveals that it does not really have a spatial dimension, since all spatial attributes are exogenously found and input to the model. No network representation is present. Similarly, classical facility-location models have only spatial attributes and typically lack the broad range of figures-of-merit typical of a siting decision. Take an obnoxious-facility model, for example, it is highly desirable to have the classic max-min objective as one of the several benefits of the DEA model, and have the cost-benefit analysis of DEA include explicitly the spatial dimension (Thomas et al. 2002). In so doing, the deviation measure is now truly a composite of both physical distance-separation and economic benefit measures. Again, a full development of this concept can be found in Chan (2005).

VIII. PRESCRIPTIVE TECHNIQUES IN LAND USE

Thus far, we have concentrated on discrete facility-location models, which lend themselves readily to prescriptive modeling. While the development is not as natural, it can be shown that prescriptive models can be constructed for land use planning as well. To show this, we will point out the linkage between the site location model of Equation 4.5 and a basic building block of land use models:

the gravity model. It will be shown that when the spatial cost function of the gravity model assumes a particular form, it becomes the site location model mentioned above. The relationship is obvious and not-so-obvious. It is quite apparent once one recognizes that the decision process to site a facility is the same as that involved in making travel plans. It is not so obvious so far as historical development is concerned, since the two models come from very different professional groups who are not familiar with each other's work until recently.

A. Entropy Maximization Model

Take the doubly constrained gravity model discussed in Chapter 3 for allocating economic activities among available land (Wilson, Coelho, MacGill, and Williams 1981). Such a model can be derived from a mathematical program of the following form:

$$\text{Max} - \Sigma_i \, \Sigma_j \, v_{ij} \ln V_{ij} \tag{4.34}$$

subject to

$$\begin{aligned} \Sigma_i \, V_{ij} &= V_i \\ \Sigma_j \, V_{ij} &= V_j \\ \Sigma_i \, \Sigma_j \, V_{ij} \, C_{ij} &= C \\ V_{ij} &\geq 0. \end{aligned} \tag{4.35}$$

As pointed out in Section XII-B of Chapter 3, this is the well-publicized Stirling approximation of the entropy maximization model in which C is the total observed travel cost expended in the trip travel pattern $\{V_{ij}\}$. The optimal solution results in a most probable distribution of activities consistent with all known information. Such information is associated with the cost and constraints placed on the interactions generated from origin i and attracted to destination j.

The Lagrangian associated with this mathematical program is

$$\begin{aligned} L\,(\mathbf{V}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, \beta) = &- \Sigma_i \, \Sigma_j \, V_{ij} \ln V_{ij} + \Sigma_i \, \alpha_i \, (V_i - \Sigma_j \, V_{ij}) \\ &+ \Sigma_j \, \gamma_j \, (V_i - \Sigma_j \, V_{ij}) + \beta \, (C - \Sigma_i \, \Sigma_j \, V_{ij} \, C_{ij}) \end{aligned} \tag{4.36}$$

and L is to be maximized over non-negative values of the trip interactions V_{ij}. The maximization problem results in non-zero values for each V_{ij} for finite values of β. The optimal solutions

$$V^*_{ij} = \exp\,[- \alpha^*_i - \gamma^*_i - \beta^* C_{ij}]$$

result, where the optimal Lagrange multipliers $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$ are shown with an asterisk (*). By defining

$$k_i = \frac{\exp\,[- \alpha^*_i\,]}{V_i}, \text{ where } l_j = \frac{\exp\,[- \gamma^*_j\,]}{V_j},$$

the doubly constrained gravity model

$$V_{ij} = k_i l_j V_i V_j F_{ij} = k_i l_j V_i V_j \ \exp\,(-\,\beta C_{ij}) \tag{4.37}$$

is obtained, as defined in Section XI-B in Chapter 3.

B. Relationship to the Allocation Model

As the parameter β tends to infinity, the model Equation 4.37 tends to the solution of the allocation model Equation 4.5 when the interactions x_{ij} are generalized from {0,1} all-or-nothing assignment to any number instead. Correspondingly, the number of assignments at i is generalized to V_i and at j to V_j (Wilson et al. 1981). Introducing the non-negativity constraint $V_{ij} \geq 0$, the appropriate first order KKT conditions for the Lagrangian Equation 4.36 are given by the dual complementary slackness [Equation 4.18], dual feasibility [Equation 4.17], and non-negativity conditions below:

$$V_{ij} \frac{\delta L}{\delta V_{ij}} = 0, \qquad \frac{\delta L}{\delta V_{ij}} \leq 0, \qquad V_{ij} \geq 0 \tag{4.38}$$

and it is necessary to consider the possibility of boundary solutions. Because $\delta L/\delta V_{ij}$ is not defined at the boundary $V_{ij} = 0$, an appropriate limiting process must be invoked to solve Equation 4.38.

To show this, consider the following Lagrangian variant of the model defined by Equations 4.34 and 4.35:

$$\text{Max}_{V_{ij}} \left\{ -\frac{1}{\beta^*} \Sigma_i \ \Sigma_j \ V_{ij} \ln V_{ij} - \Sigma_i \ \Sigma_j \ V_{ij} C_{ij} \right\} \tag{4.39}$$

subject to

$$\begin{gathered} \Sigma_j \ V_{ij} = V_i \\ \Sigma_i \ V_{ij} = V_j \\ V_{ij} \geq 0 \end{gathered} \tag{4.40}$$

for the value of β^* which satisfies the travel cost constraint in the set of constraints Equation 4.35. The optimal Lagrange multipliers $\hat{\alpha}^*_i \, \hat{\gamma}^*_i$ of the resulting Equation 4.36, $V_{ij} = \exp\,[-\beta^* \, (\hat{\alpha}^*_i + \hat{\gamma}^*_j + C_{ij})]'$ are related to those associated with the entropy maximizing model by $\beta^* \, \hat{\alpha}^* = \alpha^*$ and $\beta^* \, \hat{\gamma}^* = \gamma^*$. The model as defined by Equations 4.39 and 4.40 is now seen as a member of a family, parameterized by β^*. Variation in β^* in the entropy maximizing model corresponds to the variation of the total travel cost C. As β^* becomes very large the relative contribution of the dispersion term $\frac{1}{\beta^*} \Sigma_i \Sigma_j \ V_{ij} \ln V_{ijl}$ becomes small compared with that from interacting costs. In the limit as β^* tends to infinity, the nonlinear program becomes the linear allocation model associated with facility location. Thus it can be seen that a basic building block of land use model, the gravity model, is just a generalization of a basic building block of facility location, the linear allocation model.

C. Optimal Control Models of Spatial Interaction

All the prescriptive models discussed so far in this chapter are static models, in which the time element is explicitly absent. Spatial dynamics has become an issue of great interest in recent decades, mostly for its capacity to model the evolution of land use over time (Nijkamp and Reggiani 1992). More specifically, an optimal control model[5] based on a cumulative entropy-function is used to establish correspondence between macro/aggregate dynamic models for spatial interaction and micro/disaggregate choice models. All trips V_{ij} are time dependent in this case. In such a control problem, state variables can be defined as the number of trips generated from zone i; V_i. A control variable may be V_{ij}, suggesting the possibility of influencing spatial movements through such means as physical restraints or user-charge incentives. It can be seen also that a spatial interaction model corresponds to a logit model. This is explained in the "Activity Allocation and Derivation" chapter of Chan (2005).

A distinct advantage of the dynamic extension is its capability in examining order and chaos. A stochastic-control formulation further suggests that under the conditions of a catastrophe behavior, the stochastic disturbances do not affect evolution of the dramatic changes—an interesting result indeed. An important corollary question is whether there exist types of models capable of generating complexity in dynamic phenomena while retaining extreme simplicity in their structure. Another corollary question is whether it is possible to empirically verify the evolution through empirical time-series data. We will tackle these cogent questions in the "Chaos" and "Spatial-Temporal Information" chapters in Chan (2005).

IX. CONCLUDING REMARKS

We have reviewed in this chapter the pertinent prescriptive techniques used in facility location/land use models. To the extent possible, we try to illustrate with examples rather than formal theoretical development, leaving much of the algorithmic procedures to Appendix 4. Classic examples include the Herbert-Stevens model, the Steiner-Weber location problem, the p-median problem, and the gravity model. Within the limited space available, hopefully we have outlined the main ideas behind a variety of heuristic and analytical methods of optimization. Included in the survey is the central notion of convergence and duality that governs many of the algorithms we use and provide much of the insights we can gain from applying prescriptive models. We also begin to show the relationship between location/allocation models, location/routing models, and spatial interaction models through such formulations as the quadratic assignment problem and entropy maximization. We conclude with some of the newest techniques available. These include the DEA evaluation model, which rank orders alternative facility locations, assuming the most efficient use of each site. An optimal control formulation extends the static spatial-interaction model to yield the evolution pattern over time. Best of all, it shows the implications of judicious intervention policies. In short, this chapter paves the way for many of the topics that build upon these elementary tools.

X. EXERCISES

Self-Instructional Module: GRAPH OPTIMIZATION

(to be found on the attached CD/DVD)[6]

Graph theory has many applications in analysis. Aside from being analogous to physical networks, its elegant form can be adapted to analyze many different problems. A physical application is the analysis of flow patterns of fluids through a network of pipes. But there are many less than obvious applications. In the area of project management, for example, the Program Evaluation and Review Technique and Critical Path Method (PERT/CPM) uses graph theory to best allocate monetary and resources to meet project deadlines. Other applications include laying power or communication lines in the best way, and constructing an "optimal" computer data-base organization. This module will lead you through some basic definitions and some applications of graph theory.

After working through this module, the reader should

(a) Have a hands-on feel of how graphs can be manipulated for best performance;
(b) See some practical applications of graphs;
(c) Understand the concept of optimization.

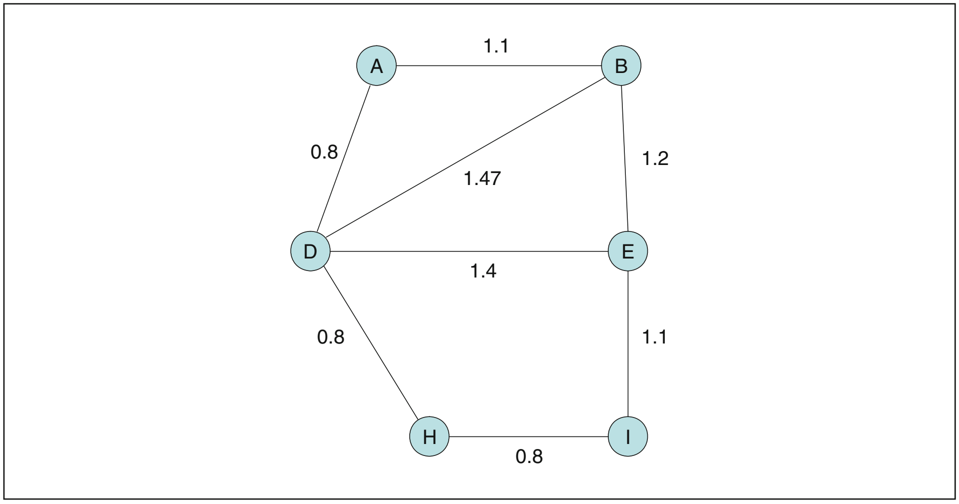
The Graph-Optimization module serves as an excellent introduction to more formal optimization methods. Using an intuitive approach, the concept of an algorithm is explained. This paves the way for more formalism. In the Integer Programming section of this chapter, for example, network optimization is formally discussed. Most important is explaining the concept of optimality. Supplemented with the Linear Programming module, the Graph-Optimization module will open the avenue for more in-depth discussion of "Optimization Schemes" in Appendix 4, which includes computational complexity analysis of algorithms.

Problem 1: Properties of a Facility-Location Model

An urban network is shown in Figure 4.18, with arc distances encoded in miles. Also shown in the table below are the demand population figures in thousands, labeled as "demand weights." Find the optimal site for locating a hospital, on the basis that the hospital should be most accessible to the entire population.

This problem is translated to finding the *absolute median* in a network, which is to locate the node with the smallest weighted sum of demands and shortest distances between the site and each demand. Let us recap the median discussion in book Section 2-VII-B. Suppose the arc travel distances between a node pair $i - j$ is d_{ij}. The demand weight at node i is f_i. As a first step, we like to normalize each nodal demand f_i by the sum of all nodal demands $\Sigma_i f_i$. It follows that the normalized demand weights, when summed over all the nodes, add up to unity. As discussed in book Sections 2-VII-A and 2-VII-B, the nodal demand weights f_i are also converted to their complements, defined as the difference between the largest demand weight and the specific nodal weight under examination. Similar to nodal demands, the complements of nodal demands are also normalized as conditional probabilities, where conditional probability is the nodal demand-complement expressed as a percentage of the total demand over all nodes.

Figure 4.18 URBAN NETWORK



SOURCE: Adapted from Handler and Mirchandani (1979)

(a) Please fill in the entries of the Table below. Based on the completed Table, please use enumeration to arrive at the absolute median.

(b) Then go through a similar discussion as in book Sections 2-VII-A and 2-VII-B for locating the median and the center.

Destination node	A	B	D	E	H	I	
Demand weights	5	2	1	0	1	3	
Normalized demand weight[7]							Weighted Sum
Complement of demand wt.[8]							
Conditional probability							
Shortest distance from node A to							
Shortest distance from node B to							
Shortest distance from node D to							
Shortest distance from node E to							
Shortest distance from node H to							
Shortest distance from node I to							

Problem 2: Maximal-Coverage Facility-Location Model

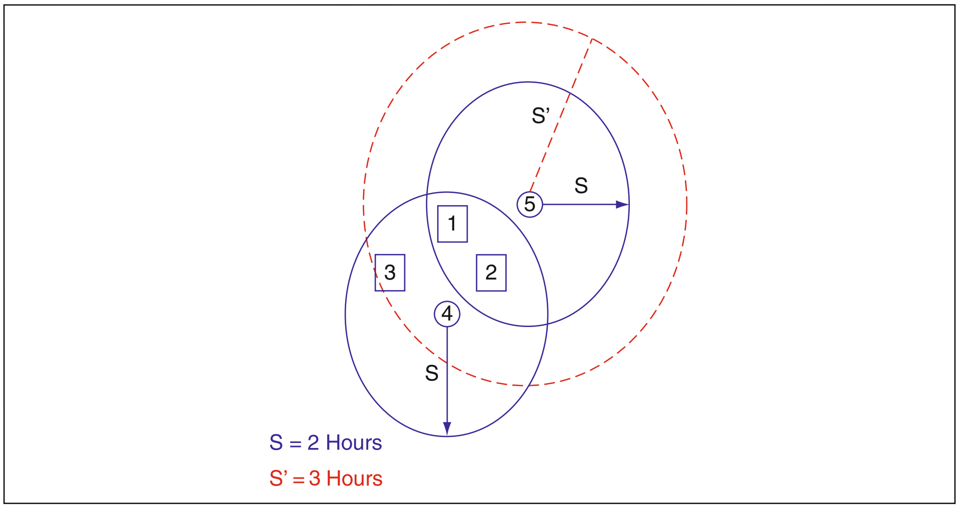
A regional airport is being contemplated. There are two demand locations (labeled as nodes 4 and 5), representing the two cities the airport is anticipated to serve. They have a population of one million and eight hundred thousand people respectively. Planners figure that passengers are willing to drive up to two hours to access an airport. Figure 4.19 has a two-hour travel-time distance drawn around each of them. Three candidate locations—labeled as nodes 1, 2, 3—are also displayed. The objective of this problem is to determine which of the three sites would best cover the demands of 10 and 8.

(a) Please formulate this problem as an integer program and solve it with the aid of a mathematical programming software such as LINGO. Trial versions of the LINGO products are available at the LINDO Systems website. While such formalism may not seem necessary for this "toy problem," solution software is absolutely necessary for larger size problems.

Using LINGO, the following solution is obtained. It can be shown by inspection that the solution makes sense.

```
MAXIMAL COVERAGE LOCATION PROBLEM
OBJECTIVE FUNCTION = 18.0
BASIC STRUCTURAL VARIABLES

X22       1.0
X44       1.0
X55       1.0

MARGINAL VALUE ANALYSIS
SHADOW PRICES FOR CONSTRAINTS

1       –10.0
2        –8.0
3        18.0

REDUCED COSTS FOR NON-BASIC STRUCTURAL
VARIABLES

X22       0.0
X33      −8.0
X41       0.0
X42       0.0
X43       0.0
X51       0.0
X52       0.0
```

Figure 4.19 EXAMPLE MAXIMUM COVERAGE PROBLEM



(b) Instead of re-running the LINGO software, can you use this output to answer the following questions? Please tell us why you can or you cannot.

- □ What happens if the demands change from 10, 8 to 5, 10 for nodes 4 and 5 respectively?
- □ What happens if the travel time radius is increased from 2 to 3 hours?
- □ What happens if we select two instead of one airport to build?
- □ Does LINGO find discrete, binary solutions automatically?

ENDNOTES

[1] Solution algorithms are explained in Section III-D below and in Appendix 4.

[2] This is in accordance with the complementary slackness conditions in LP, where a non-zero dual variable value is associated with a tight primal constraint (that is a constraint satisfied strictly at equality or land is developed 100 percent). Reversely, a dual variable is zero-valued at constraints that are strict inequalities, which is the case in point when there is surplus land. The following section will elaborate on this point.

[3] See the satellite tracking station placement examples within the last sections of the "Facility Location" chapter in Chan (2005).

[4] A root arc corresponds to an artificial variable in regular LP, making up a full rank of m in a node-arc incidence matrix of rank $m - 1$. (See Appendix 4 for more details.)

[5] For an introduction to optimal control theory, please see Appendix 1. Also consult the dynamic programming example in Appendix 3.

[6] The answer to this Module is attached at the end of this textbook.

[7] Normalized demand weight at node i is the nodal demand f_i expressed as a percentage of the total demand over all nodes.

[8] If the maximum demand at node f_i is M, the complement at all other nodes becomes $M - f_i$, where f_i is the demand at node i.

REFERENCES

Ahuja, R. K.; Magnanti, T. L.; Orlin, J. B. (1993). *Network flows: Theory, algorithms, and applications.* Englewood Cliffs, New Jersey: Prentice-Hall.

Au, T.; Stelson, T. E. (1969). *Introduction to systems engineering: Deterministic models.* Reading, Massachusetts: Addison-Wesley.

Bartholomew, A.; Brown, S.; Chan, Y. (1990). Airport location problem: Initial formulation. Working Paper. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson AFB, Ohio.

Bazaraa, M. S.; Jarvis, J. J.; Sherali, H. D. (1990). *Linear programming and network flow.* New York: Wiley.

Bersekas, D. P. (1991). *Linear network optimization: Algorithms and codes.* Cambridge, Massachusetts: MIT Press.

Brimberg, J.; Chen, R.; Chen, D. (1998). "Accelerating convergence in the Fermat-Weber location problem." *Operations Research Letters* 22:151–157.

Brooke, A.; Kendrick, D.; Meeraus, A. (1995). *GAMS: Release 2.25.* (Scientific Press Series). Boston: Boyd and Fraser.

Burkard, R. E. (1990). "Locations with spatial interactions: The quadratic assignment problem." In *Discrete facility location theory,* edited by P. Mirchandani and R. L. Francis. New York: Wiley-Interscience.

Cameron, D. M.; O'Brien, D. L.; Chan, Y. (1990). Airport location problem: Alternative formulation. Working Paper. Deptartment of Operational Sciences. Air Force Institute of Technology. Wright-Patterson AFB, Ohio.

Chan, Y. (2005). *Location, transport and land-use: Modeling spatial-temporal information*. Berlin and New York: Springer.

Chen, P. C.; Hansen, P.; Janmard, B.; Tuy, H. (1998). "Solution of the multisource Weber and conditional Weber problems by D.-C. programming." *Operations Research* 46, No. 4:548–562.

Claunch, E.; Goehring, S.; Chan, Y. (1962). Airport location problem: Three- and four-city cases. Working Paper. Deptartment of Operational Sciences. Air Force Institute of Technology. Wright-Patterson AFB, Ohio.

Devish, O.; Arentz, T.; Borger, A.; Timmermans, H. (2006). "Bilevel negotiation protocol for multiagent simulation of housing transactions and market clearing processes." *Transportation Research Record*, No. 1977:84–92.

Francis, R. L.; McGinnis Jr., L. F.; White, J. A. (1992). *Facility layout and location: An analytical approach.* Englewood Cliffs, New Jersey: Prentice Hall.

Hamerslag, R.; Van Berkum, E. C.; Repolgle, M. A. (1993). A model to predict the influence of new railways and freeways on land use development. (Presentation Paper 930875). Paper presented at the 72nd annual meeting of the Transportation Research Board, Washington, D. C.

Handler, G. Y.; Mirchandani, P. B. (1979). *Location on Networks: Theory and Algorithms.* Cambridge, Massachusetts: MIT Press.

Hansen, P.; Mladenovic, N.; Taillard, E. (1998). "Heuristic solution of the multisource Weber problem as a *p*-median problem." *Operations Research Letters* 22:55–62.

Harry, D.; Farmer, R.; Chan, Y. (1995) Airport location problem: Comparison of algorithms. Working Paper. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson AFB, Ohio.

Hillier, F. S.; Lieberman, G. J. (1986). *Introduction to operations research.* San Francisco, California: Holden-Day.

Hillier, F. S.; Lieberman, G. J. (1990). *Introduction to mathematical programming*. New York: McGraw-Hill.

Hurter, A. P.; Martinich, J. S. (1989). *Facility location and the theory of production.* Boston: Kluwer Academic Press.

Kent, B.; Bare, B. B.; Field, R. C.; Radley, G. A. (1991) "Natural resource land management planning using large-scale linear programs: The USDA forest experience with FORPLAN." *Operations Research* 39:13–27.

Lasdon, L.; Warren, A. (1986) *GINO: General interactive optimizer.* (Scientific Press Series). Femcraftvillage, Massachusetts: Boyd and Fraser.

Leonard, R.; McDaniel, P.; Nelson, R. (1991). Airport location problem. Working Paper. Deptartment of Operational Sciences. Air Force Institute of Technology. Wright-Patterson AFB, Ohio.

Magnanti, T.; Wong, R. T. (1990). "Decomposition methods for facility location problems." In *Discrete facility location theory*, edited by Mirchandani, P. and Francis, R. L. New York: Wiley-Interscience.

McEachin, R.; Taylor, G.; Chan, Y. (1992). Airport location problem: Linear and nonlinear formulations and solutions. Working Paper. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson AFB, Ohio.

Nijkamp, P.; Reggiani, A. (1992). *Interaction, evolution and chaos in space.* Berlin and New York: Springer-Verlag.

Raulerson, J.; Bowyer, R.; Zornick, J.; Chan, Y. (1994). The Herbert-Stevens residential model. Working Paper. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson AFB, Ohio.

Ravindran, A.; Philips, D. T.; Solberg, J. (1987). *Operations research: Principles and practices*, 2nd ed. New York: Wiley.

Russell, D. M.; Wang, R.; Berkhin, P. (1992). *HiQ reference manual.* Los Gatos, California: Bimillennium Corporation.

SAS Institute Inc. (1991). *Changes and enhancements to SAS/OR software,* (Release 6.07). Preliminary documentation. Cary, North Carolina.

SAS Institute Inc. (1991). *SAS/OR user's guide.* (Version 5). Cary, North Carolina.

Thomas, P.; Chan, Y.; Lehmkuhl, L.; Nixon, W. (2002). "Obnoxious-facility Location and Data-envelopment Analysis: a Combined Distance-based Formulation." *European Journal of Operational Research* 141, No. 3:495–514.

Vernon, R.; Lanning, J.; Chan, Y. (1992). The residential allocation prediction model: A linear-programming/network-flow problem. Working Paper. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson AFB, Ohio.

Wilson, A. G.; Coelho, J. D.; Macgill, S. M.; Williams, H. C. W. L. (1981). *Optimization in locational and transport analysis.* Chichester, England: Wiley.

Winston, W. L. (1991). *Operations research: Applications and algorithms,* 3rd ed. Belmont, California: Duxbury Press.

5

Multicriteria Decision Making

"I have hardly ever known a mathematician who was capable of reasoning."
Plato

While most of us have practiced **multicriteria decision making (MCDM)** in our business and personal life, it is relatively recent that the knowledge base for such a procedure has been organized and quantified into a formal set of methodologies. In some ways, it represents the amalgamation of descriptive and prescriptive models in the context of behavioral sciences. Descriptive models were defined in Chapter 3 to include such techniques as the conventional use of simulation and statistics that replicate the real world scenario. Prescriptive models, on the other hand, refer to procedures, such as optimization, which go one step further to arrive at a desirable course of action. We will show in this chapter that through the integration of both descriptive and prescriptive procedures, the role of quantitative analysis becomes clear in a pluralistic society with many interests and aspirations.

First, we try to capture the fundamental ideas behind MCDM, particularly as it is applied toward location decisions (Massam 1988). Because of page limitations, this chapter may not be as comprehensive a treatment as the excellent methodological texts such as Chankong and Haimes (1983), Goichoechea, Hansen, and Duckstein (1982), Seo and Sakawa (1988), Yu (1985), and Zelany (1982). The main ideas behind these comprehensive treatments, however, are hopefully reported in a summarized format, with liberal examples drawn from location decisions to bring alive the concepts. This document also differs from others in introducing MCDM from a prescriptive framework, rather than a descriptive framework. Unlike most books on this subject, we approach the subject from **multicriteria optimization,** while most texts start with decision analysis. An advantage of this approach is that a deterministic view of the world is adopted in the beginning. Only at a later stage do we assume knowledge of probabilistic concepts. This is judged to be more intuitive in our opinion. A second advantage is that we tend to view the MCDM process as a whole very early in the discussion. Thus the basic concepts are put immediately at the front end. While this may violate the scientific tradition of

Y. Chan, *Location Theory and Decision Analysis*, 2nd ed., DOI 10.1007/978-3-642-15663-2_5,

axiomatic development, we hope the style is more compatible with the application oriented standpoint, in which the question of "what MCDM does for me?" is answered in the first few pages. Such a way of thinking appears to be endorsed by Goichoechea et al. (1982) and Kirkwood (1997).

I. PREFERENCE STRUCTURE

We like to introduce MCDM in terms of the X, Y', and Z' space. X is often referred to as the **alternative set** or the **decision space.** Y' is the criterion set or the outcome space, and Z' is the preference structure. For example, sites A and B are military bases that are candidates for closure. A decision is to be made to close one of the two. Thus in the X-space, we defined a vector of two binary variables $\mathbf{x} = (x_1, x_2)$, where the binary variables take on the unitary value when base A and B are closed, and zero otherwise. The impact of a base closure may be measured in terms of a peace dividend and the defense posture in case of war, as represented by the two entries of the vector $f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))$ respectively. This criteria set can be written compactly as $\mathbf{f} = (y'_1, y'_2)$, where y'_1 may be quantified in terms of dollar cost savings in peace time, and y'_2 is readiness in case of war, measured, for instance, by the number of emergencies reachable within a travel radius (i.e., geographic coverage in hours of flight time). The vector $\mathbf{y} = (y'_1, y'_2)$ in this case is defined in the Y' space. Out of the y_1s, the alternative with the best figure of merit $\mathbf{y}'^*$ is picked by the decision maker(s) according to some preference structure Z'. Let us say that it corresponds to alternative $\mathbf{x}^*$, which is the vector (1, 0) when $\mathbf{y}'^*$ is mapped back to the X space via the inverse function $\mathbf{x}^* = f^{-1}(\mathbf{y}'^*)$, indicating that base A, rather than B, should be closed. All MCDM problems evolve around such mapping between X, Y' and Z' space. We hasten to add that while the mapping between X and Y' is relatively straightforward, the mapping between Y' and Z' (or the preference structure), tends to tax the limitation of the state of the art.

A. The Importance of Preference Structure

Quantifying the preference structure is the most taxing part of MCDM. It is in fact the heart and soul of the modeling procedure. To illustrate the Z space, let us elaborate on the above example and further suggest that of the two bases slated for closure, A is a scientific base while B is a tactical base, meaning that B is closely linked to combat while A is somewhat removed from it. Let us further consider the decisions in the context of two different philosophical outlooks: pessimistic and optimistic. Here we show the payoffs of the two alternative decisions—or utility measures by which we have put cost savings and geographic coverage in the same scale. The common scale ranges in this case from 0 (worst) to 100 (best):

	Criteria set f:			
	Utility if there is			
	peace	war	Min payoff	Max payoff
	$f_1(\mathbf{x})$	$f_2(\mathbf{x})$		$\mathbf{y}'$
Alternative Set X				
base A closure ($x_1 = 1$)	50	90	(50)	(90)
base B closure ($x_2 = 1$)	100	30	(30)	(100)
	↑ Y'			

The data show that if base *A* is closed, it will look ill-advised (50 points in a 100) if peace is maintained. The reason is that the scientific know-how generated from base *A*, aside from being a long-term defense investment, could have spinoffs in the civilian economy. In conducting a war on a real-time basis, however, it is much more astute to keep the tactical base *B* while scientific base *A* appears to have little bearing upon the day-to-day fighting. Considering these tradeoffs, the question is again: Should *A* or *B* be closed—assuming only one of the two bases is to be closed?

A pessimistic decision maker will plan for the worst and will look at the worst possible outcome corresponding to each of the two decisions. This is between closing *A* or *B*, with a minimum payoff of 50 and 30 respectively. In other words, should *A* be closed, the decision maker wants to anticipate the worst result or the minimum payoffs that correspond to a peace outcome. Should *B* be closed, on the other hand, the decision maker would like to plan for the war outcome. On the other hand, an optimistic decision maker will plan for the best possible outcomes or maximum payoffs, war if *A* is closed and peace if *B* is closed, corresponding to scores of 90 and 100 respectively. Both decision makers are trying to make the best of the situation, or maximizing the respective payoffs. The pessimist (who is going to maximize the minimum payoffs) will close *A* while the optimist (who is going to maximize the maximum payoffs) will close *B*. In other words, based on the same set of data, different preference structures will lead toward an entirely different decision.

Suppose now we have three locations to consider—*A, B,* and *C*—instead of two. Let us rank them in order of preference. In a general case, the ranking can be compiled for base closing, for warehouse location, or for siting of a manufacturing plant. Let us say PQR Corporation is considering three states for building a new manufacturing plant. The candidate sites in States 1, 2, or 3 are evaluated by three criteria (Yu 1985):

	Scores		
Criteria	$\mathbf{y}^1$	$\mathbf{y}^2$	$\mathbf{y}^3$
Labor force	6	7	8
Transport	7	8	6
Tax breaks	8	6	7

where each score y_{ij} is between 0 and 10, with 10 being the most desirable. Each score is specified for candidate site *j* on criterion *i*. For example, on labor-force availability, State 3 ranks the highest, State 2 ranks second, and State 1 is last. Here $Y' = \{\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3\}$, with the score for each State $\mathbf{y}^1$ $(6, 7, 8)^T$, $\mathbf{y}^2 = (7, 8, 6)^T$ and $\mathbf{y}^3 = (8, 6, 7)^T$.

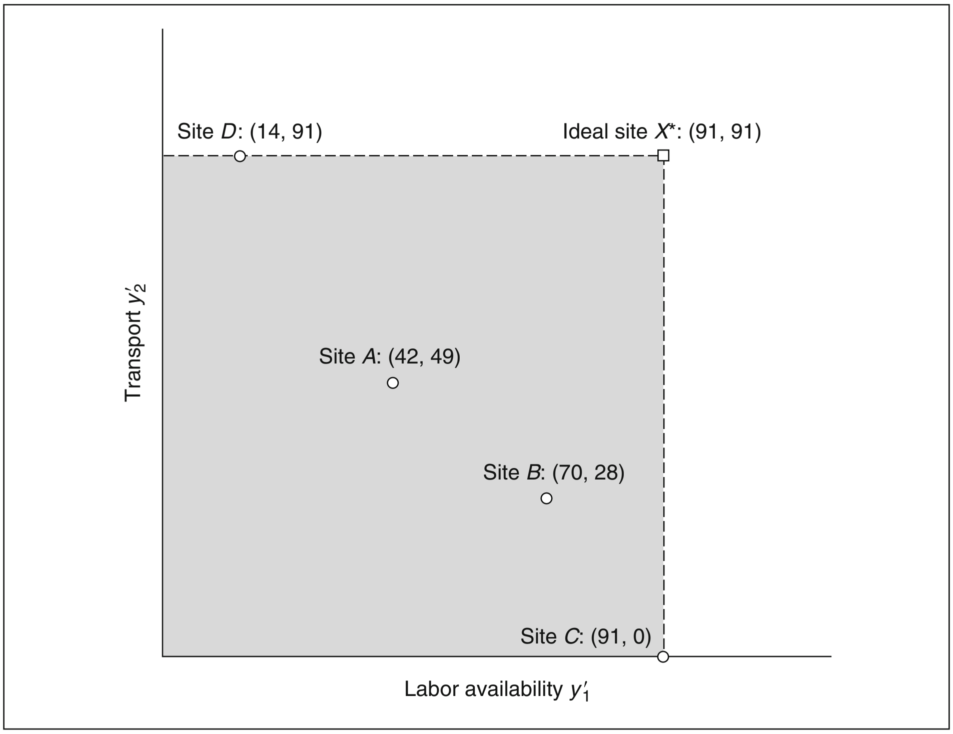
Now suppose PQR Corporation adopts the following preference structure arbitrarily. It decides that a State is preferred to another if at least in one out of three criteria, the State is leading by 2 or more points, and in the remaining two criteria, the State is not inferior to the others by more than 1 point. Put it in another way, State 1 is preferred to State 2, or $\mathbf{y}^1 > \mathbf{y}^2$, if and only if State 1 offers at least one criterion *i* in which $y^1_i - y^2_{\mathrm{i}} \geq 2$ and $y^2_{\mathrm{k}} - y^1_{\mathrm{k}} < 2$ for the remaining criteria $\{k \neq i\}$. One can now picture the resulting ranking as a cyclic relationship: $\mathbf{y}^1 > \mathbf{y}^2$, $\mathbf{y}^2 > \mathbf{y}^3$ and $\mathbf{y}^3 > \mathbf{y}^1$ (instead of $\mathbf{y}^1 > \mathbf{y}^3$ as one would expect from transitivity)[1]. Aside from non-transitivity, this example emphasizes the important role preference structure plays in ranking alternatives. It is conceivable that if another preference structure be adopted, transitivity may result.

B. Paired versus Simultaneous Comparison

Instead of a pairwise comparison between alternatives, another way is to simultaneously compare the alternatives against an ideal, where an ideal point has the best components in the outcome space. For example, one wishes to rank the four alternatives pictured in Figure 5.1, where the two-criteria measurements of an alternative—labor availability and transport—are displayed. Thus site *A* has a labor force availability rating of 42 out of a best of 100 and a transportation index of 49. The ideal score is 91 for both labor availability and transport defining an ideal alternative X^* of (91, 91). If preferences are measured in terms of the Euclidean beeline distance displacement from the ideal, then the order of preference is $A > B > D > C$ as one can visually inspect from Figure 5.1. Here ranking is obtained by a simultaneous comparison among all alternatives (Zelany 1982).

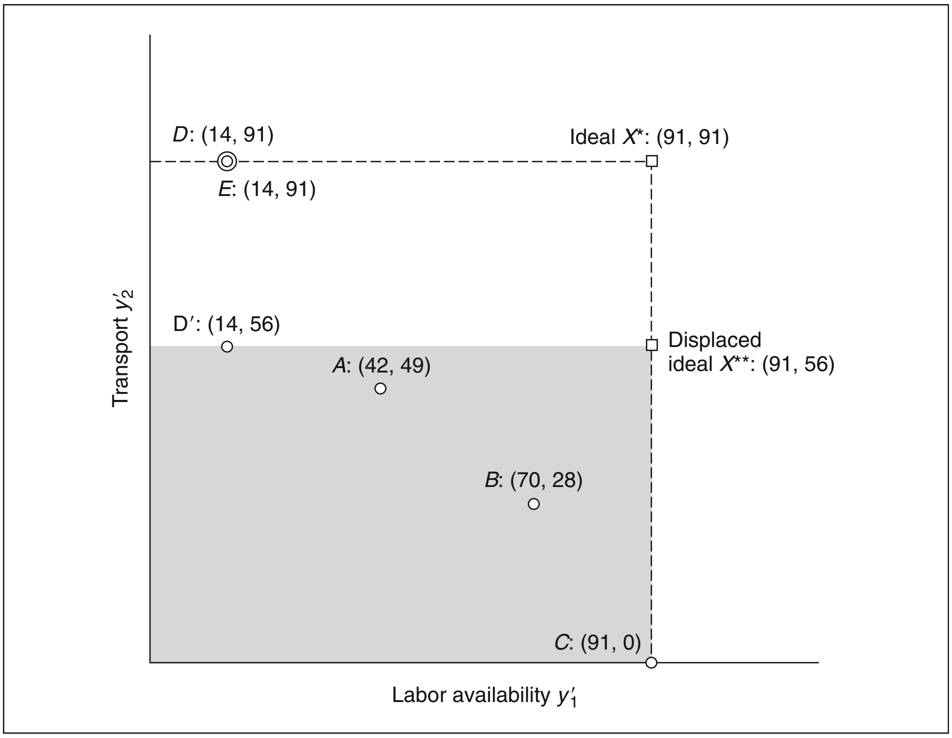
Assume that an increase in shipping rates has caused a shift from *D*: (14, 91) to *D′*:(14, 56), or transportation is now no longer as convenient as before. This shifts the ideal point X^* to (91, 56), as shown in Figure 5.2. The new ranking now—again based on Euclidean-distance displacement—is $B > A > C > D'$. Notice the preference ranking between *A* and *B* has been reversed as a result, as has *C* and *D′*. Suppose another site with the same characteristics as the original site *D* is now being considered. Call this site *E*. Introducing site *E*:(14, 91) into the

Figure 5.1 IDEAL SITE AND RANKING AMONG LOCATIONS



SOURCE: Zelany (1982). Reprinted with permission.

Figure 5.2 DISPLACEMENT OF THE IDEAL

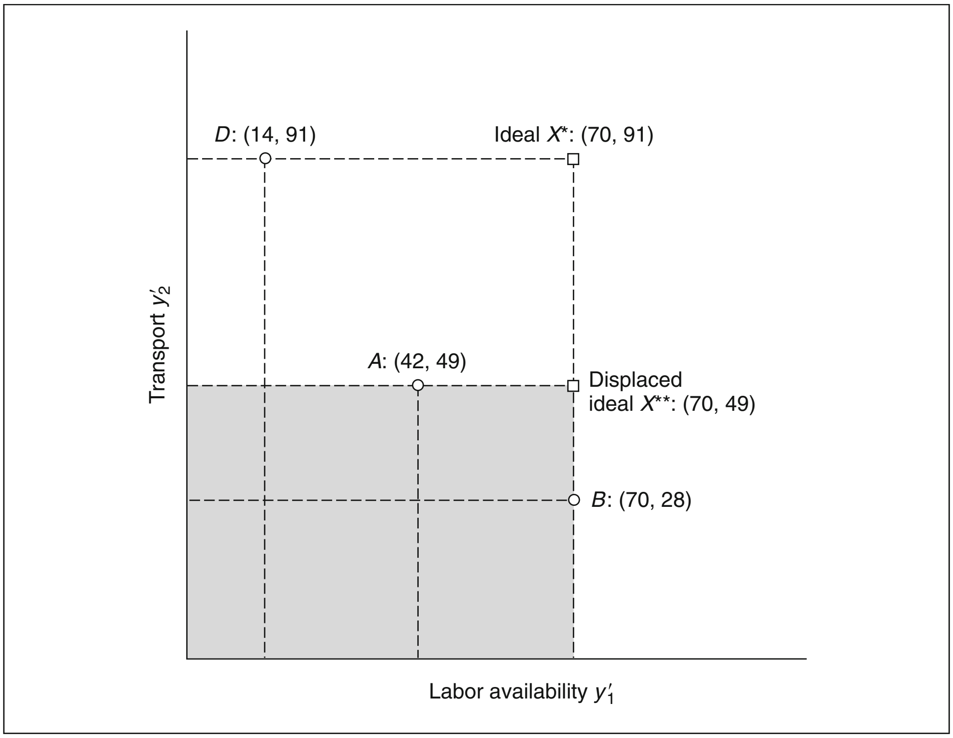


SOURCE: Zelany (1982). Reprinted with permission.

candidate set changes the ranking to $A > B > E > D' > C$. A nonoptimal alternative A has been made optimal by adding E, the replacement for D, back to the feasible set X. This is somewhat contradictory to traditional decision analysis, which assumes a definite utility associated with each alternative, and the alternative with a higher utility is always preferred.

Instead of continuing with a simultaneous comparison among the alternatives, we will now return to paired comparisons, just to show another point. In comparing A with B, the decision maker uses X^* as a point of reference. A is compared with X^* and B is compared with X^*, each separately. The comparison between A and B is an indirect consequence of this process. Now consider the very first case once more. We shall explore a particular triad of options, say $\{A, B, D\}$, shown in Figure 5.3. After comparing between B and D, one concludes that $D > B$. Next, observe that between A and D, $A > D$. Now that D has been exhaustively compared with A and B, it can be discarded from consideration ("out of sight, out of mind"). This leads to the pairwise comparison between A and B, resulting in $B > A$ because the removal of D has induced the displacement of X^* to X^{**}. The net result is: $A > D$, $D > B$, and $B > A$, which—similar to the plant-location example—is again not transitive. If we start the process with the paired comparison between A-D, it will result in $A > D$. We then establish that $A > B$ by comparing A and B, and finally $D > B$, resulting in a fully transitive relationship $A > D > B$. This shows the order we ask questions in a survey can bias the results.

Figure 5.3 SEQUENTIAL CHOICE AMONG THREE ALTERNATIVES



SOURCE: Zelany (1982). Reprinted with permission.

It can be seen that, depending on the sequential process assessing preference among alternatives, the resulting ordering is different. In situations involving sequential displacements of the ideal, the order of preference can be changed significantly. There is also a possibility of intransitivity, when a sequence of pairwise comparisons are performed. Thus the preference structure, which is often represented by the utility or value function $v(\mathbf{y}')$, is in fact the most difficult procedure in MCDM. Depending on the way one valuates the outcomes, such as comparing them against a current ideal or viewing them in an optimistic or pessimistic light, the ranking of alternatives can be very different.

II. SIMPLE ORDERING

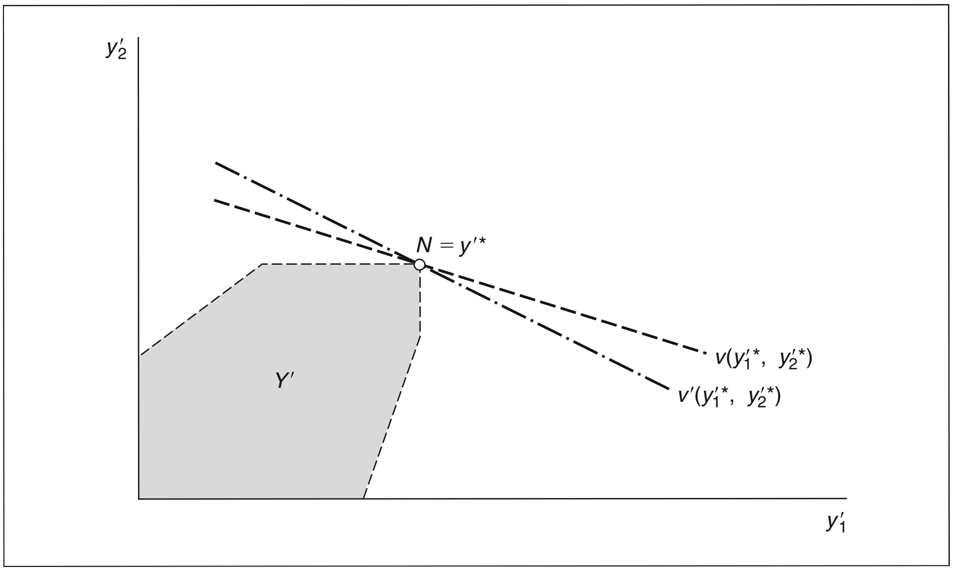
Given the difficulties with preference structures, are there any guidelines and procedures ready for making decisions? The answer is a resounding yes, and that is where a discussion such as the current one may become useful to the readers, not only in giving warnings about the danger of performing analyses incorrectly, but also in providing guidelines for the correct procedures.

The most straightforward case of MCDM is simple ordering among alternatives, where no preference structure is required. In other words, we work strictly in the X and Y' space, requiring no resolution among incompatible criteria in the Z space. In other words, we dispense with the "apples versus oranges" tradeoff. It also avoids the pitfalls associated with many surveys based on paired comparisons in which the order in which questions are asked results in very different rankings among alternatives, as we pointed out earlier. It turns out that using simple ordering, one can readily rule out a large percentage of the alternatives, leaving only very few to consider. As such, this is a very useful analysis technique on many occasions.

Assuming that we all agree upon "the more the merrier" philosophy, the concept of dominance then comes in naturally as a simple ordering tool. In the above example, if site *A* is better than site *B* in all the criteria: labor availability, transportation and tax breaks, few would disagree with the choice of *A* over *B*. Such a dominance relationship among alternatives is often referred to as **Pareto preference.** An outcome $\mathbf{y}'$ is said to be Pareto optimal if and only if it is a non-dominated solution (or it is an *N*-point). A Pareto optimal solution is also called an **efficient, noninferior, non-dominated,** or **admissible solution.** For example, the ideal solutions illustrated in Figure 5.3 and Figure 5.1 are nondominated solutions because they are equal or better than all the other alternatives in the two criteria considered: labor availability and transportation. Straightforward as it may appear, the concept of dominance is a rather robust tool. The set of *N*-points offers preference determination under conditions of ignorance (about how to compare incommensurate attributes such as y'_1 and y'_2). Thus there is no requirement to make difficult tradeoffs. In Figure 5.1, Figure 5.2, and Figure 5.3, for example, regardless of the exact value of utility/value function definition, $v(y'_1, y'_2)$, its maximum will be the *N*-point X*: $v^*(y'^*_1, y'^*_2) \geq v_i(y^i_1, y^i_2)$, where $y'^*_1 \geq y^i_1$ and $y'^*_2 \geq y^i_2$.

It makes sense to explore Y' and characterize its set of *N*-points before engaging in the assessment of v. It is possible an alternative will emerge such as shown in Figure 5.4, which is often known as the conflict-free solution. Put it in the context of Figure 5.3, Figure 5.2, and Figure 5.1, such a "win-win" solution represents the "ideal." Even though such situations seldom arise, it is well-advised to recognize them when they appear, since it saves a good deal of negotiation and hard work.

Figure 5.4 CONFLICT-FREE SOLUTION



III. EXPLORING THE EFFICIENT FRONTIER

Perhaps one of the best ways to explore the efficient frontier and to illustrate the *X*, *Y′* and *Z′* space of MCDM is still through a **multicriteria linear program** MCLP. Consider this example (Yu 1985): A corporation is deciding how much company housing to provide the employees (x_1) and the amount of 'commercial housing' from the free market (x_2)—with the amount of housing measured in square footage. The corporation in this case has to balance between welfare to the employees f_1 and the corporate goal f_2—both of which the company wants to –maximize: Max $f_1(x_1, x_2) = 4x_1 - x_2$ and Max $f_2(x_1, x_2) = 2x_1 + 5x_2$. Here, the first criterion is the employee welfare, which is enhanced by the availability of company housing, since it is cheaper to the employees than commercial housing. The second is the corporate goal, which refers to the bottom line, wherein commercial housing is less expensive on the company's pocketbook. Obviously the two criteria are not necessarily congruous and hard decisions have to be made regarding the tradeoffs between these incompatible objectives.

Now only a certain amount of housing subsidy is obligated by the company to each employee for use in both company and commercial housing. A company and commercial dwelling unit has different costs associated with them, with commercial housing one and a half times more expensive for the family pocketbook: $2x_1 + 3x_2 \leq 12$. Only a certain amount of commercial housing is available within commuting distance from the company: $x_2 \leq 3$. It is corporate policy to at least provide an amount of company housing somewhat commensurate with the commercial housing available—at one third of the footage: $3x_1 - x_2 \geq 0$. Finally, nonnegativity applies to the decision variables since they represent footage of housing: $x_1, x_2 \geq 0$.

In order to solve this multicriteria LP, a proven method is to convert one of the two criterion functions, say $f_2(x)$ into a constraint, which is added to the existing constraint set $x \in X$:

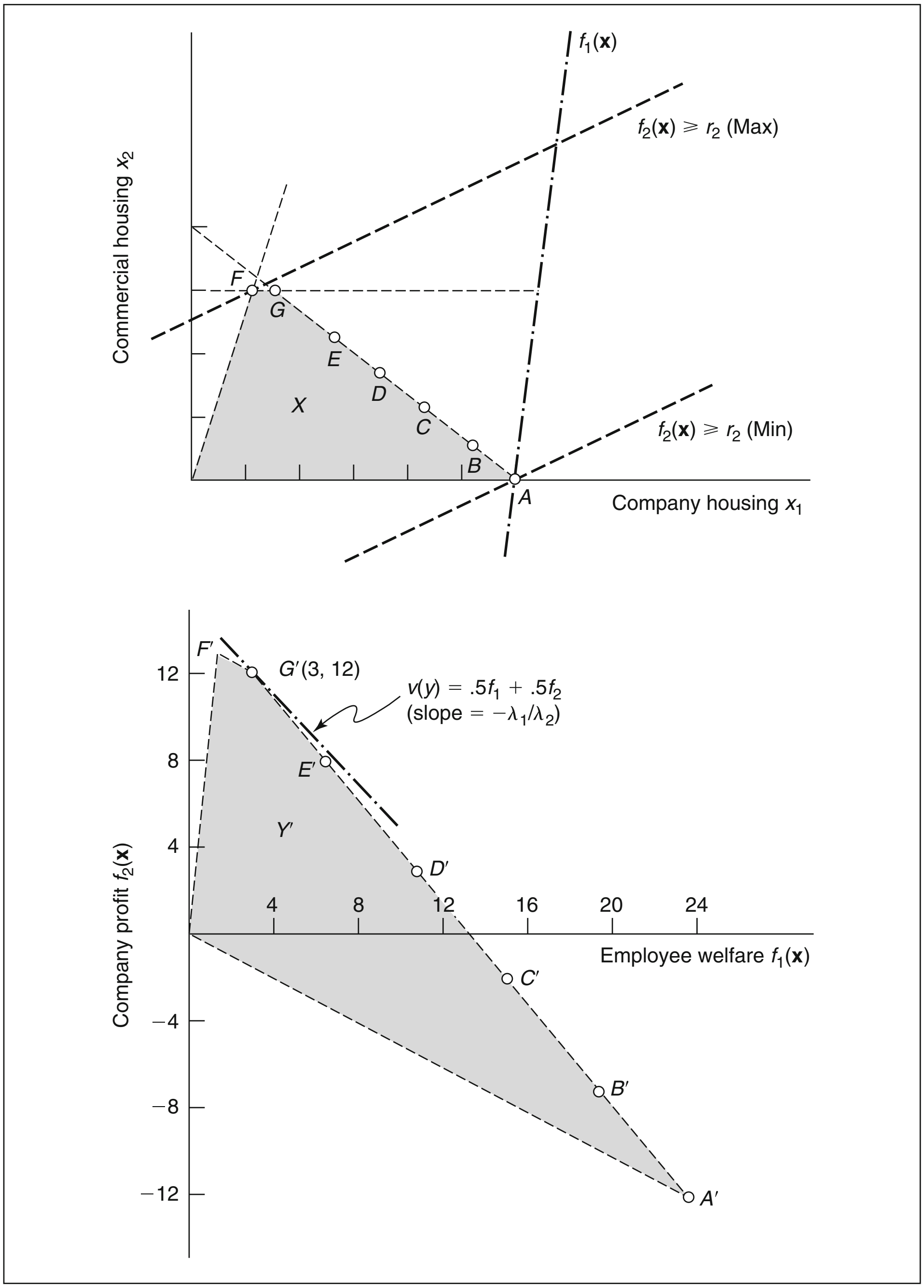
$$\begin{gathered}\text{Max } f_1(x_1, x_2) = 4x_1 - x_2 \\ \text{s.t. } x \in X, \\ f_2(x_1, x_2) = -2x_1 + 5x_2 \geq r_2\end{gathered}$$

where r_2 is a satisficing level for f_2, say the acceptable company profit. By graphically minimizing and maximizing f_2 over *X*, the feasible region defined by the original constraint set, we find $-12 \leq f_2(x) \leq 13$. Solving the above LP with r_2 varying from −12 to 13, we can find all *N*-points:

r_2	(x_1, x_2)	(f_1, f_2)
−12	A (6, 0)	A′ (24, −12)
−7	B (5.06, 0.63)	B′ (19.63, −7)
−2	C (4.13, 1.25)	C′ (15.25, −2)
3	D (3.19, 1.88)	D′ (10.88, 3)
8	E (2.25, 2.50)	E′ (6.50, 8)
12	G (1.50, 3)	G′ (3, 12)
13	F′ (1, 3)	F′ (1, 13)

as illustrated in the X-space (Figure 5.5) and the Y'-space. Notice by virtue of the computation shown in tabular form above, the (f_1, f_2) column and the corresponding points in the figure, A–F in the X-space and A'–F' in the Y-space constitutes an efficient frontier which dominates over all points $y' \in Y'$. Such an

Figure 5.5 THE DECISION AND OUTCOME SPACE



SOURCE: Yu (1985). Reprinted with permission.

N-set is generated by the **constraint reduced feasible region method,** where one of the two criterion functions has been eliminated by converting it into a constraint (Steuer 1986).

A logical question to ask at this point is: What (λ_1', λ_2') would make $A', B', \ldots,$ or F' a maximum point of $v(\mathbf{f}) = \lambda_1' f_1' + \lambda_2' f_2'$ over Y'? Now entries in the first row of the LP tableau is $-(4\lambda_1' - 2\lambda_2')$ under x_1 and $-(-\lambda_1' + 5\lambda_2')$ under x_2. Consider the existing maximum $\mathbf{G}'$ for instance. All λ's satisfying $\boldsymbol{\lambda}'(\mathbf{y}' - \mathbf{G}') \geq 0$, $\mathbf{y}' \in Y'$, or $(\lambda_1, \lambda_2)\begin{bmatrix} y_1-3 \\ y_2-12 \end{bmatrix} \geq 0$, can—among other conditions—shift the maximum point of $\lambda'\mathbf{f}$, or $(\lambda_1', \lambda_2')\begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$ to other N-points. Here, the slope of the objective function $\lambda.\mathbf{f}$ is $-\lambda_1/\lambda_2$ and the gradient is λ_2/λ_1. The inequality $\lambda(\mathrm{y}-\mathrm{G}) \leq 0$ places a stipulation on both the slope and the gradient. Specifically, the weights in $\boldsymbol{\lambda}'$ must satisfy $(y_2' - 12)/(y_1' - 3) \leq -\lambda_1'/\lambda_2'$, to cause a shift, where the **y**'s are the points within the feasible region Y', or the set of feasible outcomes. Take $F'(1, 13)$ in Y. We have $(13-12)/(1-3) \geq -\lambda_1'/\lambda_2'$. This means the slope $-\lambda_1'/\lambda_2'$, must be less than or equal to $-1/2$, or at the minimum $\lambda_1'/\lambda_2' = 1/2$. One can see the similarity between this and the optimality test used in regular LP.

Let us denote these conditions as the **weight cone** $\Lambda(G')$. The table below illustrates examples of $\Lambda(G)$ and $\Lambda(F)$, given $\lambda_1' + \lambda_2' = 1$ by convention. Take another example point A'(24 −12). We write $(-12 - 12)/(24 - 3) \leq -\lambda_1'/\lambda_2'$, or $\lambda_1'/\lambda_2' \leq 8/7$, for exploring the right of G'. This means $\lambda_1' = 8/15$ and $\lambda_2' = 7/15$.

λ_1'	λ_2'	$\mathbf{y}'^*$	f_1^*	f_2^*
. . .	. . .	. . .	. . .	. . .
0.5	0.5	G'	3	12
0.333	0.667	F′ – G′	1–3	12–13
0	1	F′	1	13

Another way of thinking about this concept is to examine a linear, additive value-function $v(\mathbf{f}) = \lambda_1' f_1 + \lambda_2' f_2$, which is rotated with a pivot at G', the table above simply records the "break points" at which an extreme point such as G' is no longer the "optimal" with regard to the value function under consideration, as the weights in the value function change. One can think of this tradeoff—or some kind of sensitivity analysis—between λ_1' and λ_2' taking place in the Z'-space.

In the context of this housing example, the multicriteria LP (or MCLP) sketches out the entire efficient frontier consisting of the illustrative points A' through G', which are members of the N-set. Irrespective of the valuation placed upon employee welfare vis-a-vis company profit, the non-dominated solutions are the only ones worthy of further examination. When more weight is placed upon the employee's welfare, a solution such as A' will make sense, which when translated back to the decision space X, means that the company would provide all employee housing. On the other hand, when company profit is valued over employee welfare, F' and F will become the viable solution, and commercial housing will provide the bulk of the living quarters for the employees. A solution such as *F*, for instance, shows 3 units of commercial housing and one unit of company housing. The criteria $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$, which are as outcomes y'_1 and y'_2, are shown as A' and F' which are (24, −12) and (1, 13) respectively. Thus the company suffers a loss if it places too much emphasis on employee welfare. On the other hand, it makes a large profit if the bottom line is closely guarded.

IV. MULTICRITERIA SIMPLEX (MC-SIMPLEX)

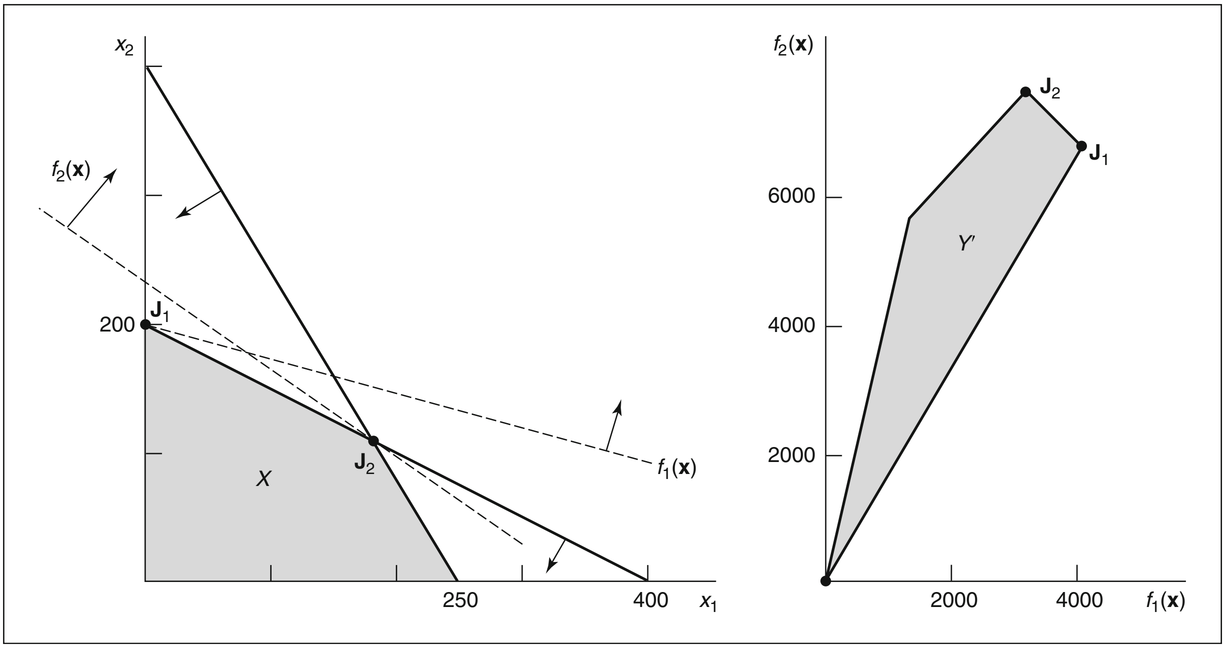
While the above example illustrates the basic concepts behind MCLP, formalization of the above problem is necessary for solving realistic size problems. Before we provide a solution algorithm, some terms need to be defined: $X = \{\mathbf{x} \in R^n | \bar{\mathbf{A}}\,\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$, where $\bar{\mathbf{A}}$ is a matrix of order $m \times n$. In other words, **x** vectors are n-dimensional nonnegative real numbers within the region defined by the constraints ${}^{A}x \leq \mathbf{b}$. Let **C** be a q by n matrix with its kth row denoted by $\mathbf{c}^k$ so that $c^k\mathbf{x}$, $k = 1, \ldots, q$, is the kth criterion function. The criteria space is thus given by $Y' = \{\mathbf{Cx} \mid \mathbf{x} \in X\}$. The objective function of the mathematical program now looks like Max $z = \boldsymbol{\lambda}^T\mathrm{C}\mathbf{x}$, which is sometimes referred to as vector optimization. Thus in the previous example, $\bar{\mathbf{A}} = \begin{bmatrix} 2 & 3 \\ 0 & 1 \\ -3 & 1 \end{bmatrix}$, $\mathbf{b} = (12, 3, 0)^T$, $\mathbf{C} = \begin{bmatrix} 4 & -3 \\ -2 & 5 \end{bmatrix}$ and $z = \lambda'_1(4x_1 - x_2) + \lambda'_2(-2x_1 + 5x_2)$. Indeed, many MCLP's have been solved using such a combined objective function z, which in effect assumes an additive-linear-value function $v(\mathbf{f}) = \lambda'_1 f_1 + \lambda'_2$. By changing the weights λ', the efficient frontier is sketched out, as already alluded to above. Such a **weighted-sum method** has its intuitive appeal, and can be made operational quite readily in many analysis offices inasmuch as it needs only a regular LP computer code. Its generality, however, is more questionable, since it may miss efficient solutions in integer-programming problems where decision variables are required to be discrete—a point we will come back to in sequel.

A. The MC-Simplex Algorithm

To formally solve an MCLP, we need an MC-simplex algorithm. For the purpose of this discussion, the best is to illustrate the basic concepts of this algorithm through a numerical example, and then refer the reader to some software that can take care of the computation on a day-to-day level. Consider the following MCLP (Zelany 1982):

$$\begin{gathered}\text{Max } f_1(\mathbf{x}) = 5x_1 + 20x_2 \\ \text{Max } f_2(\mathbf{x}) = 23x_1 + 32x_2 \\ \text{s.t. } 10x_1 + 6x_2 \leq 2500 \\ 5x_1 + 10x_2 \leq 2000.\end{gathered}$$

Figure 5.6 GRAPHICAL SOLUTION OF A MULTICRITERIA LINEAR PROGRAM IN X AND Y' SPACE



with the normal nonnegativity constraints on the decision variables. For comparison and illustration purposes, graphical solution to the problem is given in Figure 5.6 which will be explained via the following algebraic procedures.

As a readily operational procedure, we can form a tableau corresponding to $z = \lambda'^T\mathbf{f}$, or that we solve the LP starting out with the combined objective function for a particular value of λ', with $0 \leq \lambda_i \leq 1$. By working with the following tableau:

current basis $\mathbf{J}_k$	x_1	x_2	x_3	x_4	RHS
	$-5\lambda'_1 - 23\lambda'_2$	$-20\lambda'_1 - 32\lambda'_2$	0	0	0
x_3	10	6	1	0	2500
x_4	5	10	0	1	2000

It can be seen from the combined $(z_j - c_j)$s in the first row why this is called the weighted-sum approach.

Now instead of solving this as a single objective LP, the MC-simplex calls for writing the first line as two lines and carrying out the simplex procedure, say the primal simplex. Thus the first line of the tableau now looks like:

criterion	−5	−20	0	0	0
rows	−23	−32	0	0	0

To be *really* organized, we should organize the tableau as follows:

y_1	y_2	x_1	x_2	x_3	x_4	RHS
0	0	10	6	1	0	2,500
0	0	5	10	0	1	2,000
1	0	−5	−20	0	0	0
0	1	−23	−32	0	0	0

We are at a position to carry out the pivoting procedures.[2] Since the second variable x_2 will benefit both of the criterion functions more so than x_1, it is introduced into the basis. Notice this is an example of **dominance:** x_2 dominates over x_1. The result of this pivot is shown below:

criterion	5	0	0	2	4000
rows	−7	0	0	$3\frac{1}{5}$	6400
x_3	7	0	1	$-\frac{3}{5}$	1300
x_2	$\frac{1}{2}$	1	0	$\frac{1}{10}$	200

It can be seen that the entry rules of a regular simplex are modified to include dominance. As another example, if the criterion rows look like

criterion	0	0	0	2	4,000
rows	−7	0	0	$3\frac{1}{5}$	6,400

instead of the above, introducing x_1 into the basis—thus effecting an alternate solution—would not hurt the first criterion function, but will improve the second by 7 per unit of increase in x_1. This implies that the existing solution with respect to the first criterion function is a dominated solution—that it is inferior to the alternate solution with x_1 in the basis. We call this alternate solution an efficient solution *N*-point, or Pareto optimum. In fact, the definition of a Pareto optimum is that it gains at the expense of nobody else.

Now back to the original problem, the solution after the first pivot is an *N*-point $\mathbf{J}_1$, in that one of the criteria rows, the $(z_j - c_j)$s for $f_1(\mathbf{x})$, are all nonnegative; meaning that on the first criterion, the solution $\mathbf{J}_1$ dominates. Continuing the simplex, the other *N*-point obtained is:

criterion	0	0	−5/7	17/7	21500/7
rows	0	0	1	13/5	7700
x_1	1	0	1/7	−3/35	1300/7
x_2	0	1	−1/14	1/7	750/7

where the second criterion function, $f_2(\mathbf{x})$, is optimized, at the expense of the first. This example contrasts nicely with the made-up criteria rows as shown above where *both* criteria can be improved by another pivot—or at least not degraded by the pivot. It can be shown that any additional pivot performed on the tableau corresponding to the second N-point will revert back to the tableau corresponding to the first N-point. Thus we have established all N-points, defined once again as the extreme points where at least one of the criteria is optimized, for a given minimum value of the other criterion, r_k. The collection of N-points have the property of dominating over all other points in Y.

If there are two possible pivot columns to choose from, then the concept of dominance again comes into play—the column that will improve $f_k(\text{x})$ the most and degrade $f_k'(\mathbf{x})$ the least ($k \neq k'$) is the one to use. For example, consider the following criteria rows:

criterion	5	0	0	6	4000
rows	−7	0	0	$-3\frac{1}{5}$	6400

It is clear that x_1, rather than x_4, should be pivoted in, since the introduction of x_1 will degrade $f_1(\mathbf{x})$ the least, but will improve $f_2(\mathbf{x})$ the most. On the other hand, the following criteria rows show a 'tie' between x_1' and x_4 as the pivoting column, inasmuch as x_1 will benefit $f_2(\mathbf{x})$ at the expense of $f_1(\mathbf{x})$ while x_4 will upgrade $f_1(\mathbf{x})$ at the expense of $f_2(\mathbf{x})$:

criterion	5	0	0	−2	4000
rows	−7	0	0	$3\frac{1}{5}$	6400

This latter example is similar to finding the $\mathbf{J}_1$–$\mathbf{J}_2$, Pareto optimum in the original example.

Referring back to the first pivot tableau containing the combined objective-function, it can be seen that solution $\mathbf{J}_1$ or $\mathbf{x}^1$ will stay optimal if the z_j-c_j entries remain nonnegative—the typical optimality condition for simplex tableaux:

$$z_1(\lambda') = 5\lambda_1' - 7\lambda_2' \geq 0$$
$$z_2(\lambda') = 2\lambda_1' + (3-{}^1/_5)\lambda'2 \geq 0$$

Alternatively,

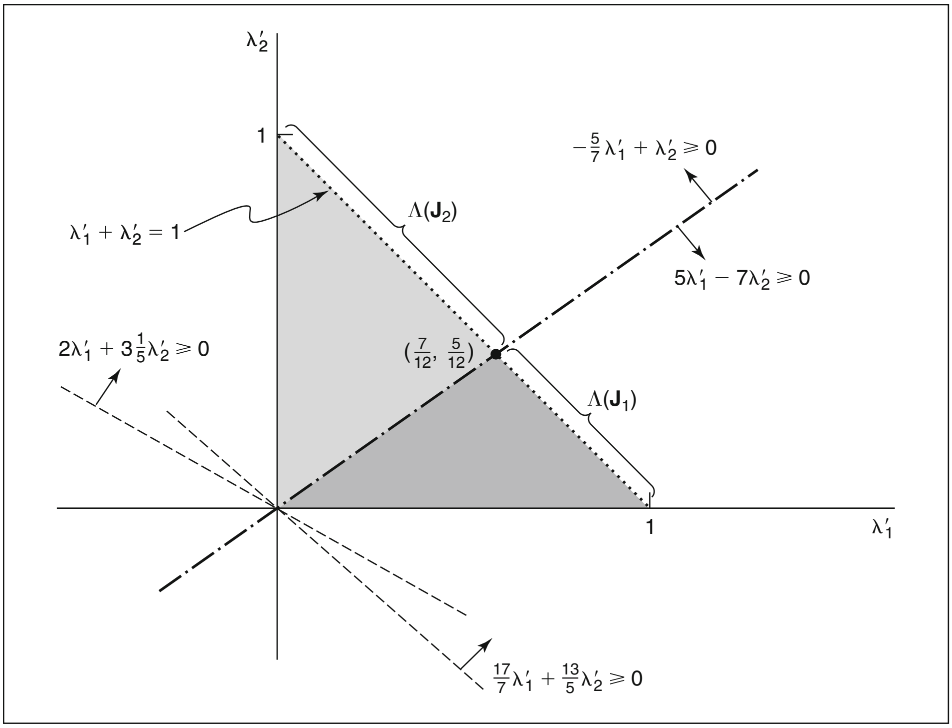
$$\Lambda(\mathbf{J}_1) = \{\boldsymbol{\lambda}' \,|\, \boldsymbol{\lambda}'^T \begin{bmatrix} 5\ 0\ 0\ 2 \\ -7\ 0\ 0\ 3\frac{1}{5} \end{bmatrix} \geq \mathbf{0}\}.$$

Similarly, we can write

$$\Lambda(\mathbf{J}_2) = \{\boldsymbol{\lambda}' \,|\, \boldsymbol{\lambda}'^T \begin{bmatrix} 0 & 0 & -\frac{5}{7} & \frac{17}{7} \\ 0 & 0 & 1 & \frac{13}{5} \end{bmatrix} \geq \mathbf{0}\}.$$

It is often convenient to display these two conditions graphically. Figure 5.7 shows the plot of the λ-space (or what we have been referred to as

Figure 5.7 THE Z'-SPACE CONTAINING THE WEIGHT CONES



the Z'-space). This figure illustrates a few cogent points. For example, it clearly shows that there is at least one point $\lambda' = (7/12, 5/12)^T$ common to both $\Lambda(\mathbf{J}_1)$ and $\Lambda(\mathbf{J}_2)$, where z reaches its maximum at both x_1 and x_2. With this λ',

$$\begin{aligned} z(\mathbf{x}^1) &= (5\lambda'_1 + 23\lambda'_2)x_1 + (20\lambda'_1 + 32\lambda'_2)x_2 \\ &= [5(7/12) + 23(5/12)]0 + [20(7/12) + 32(5/12)]200 = 5000. \end{aligned}$$

Similarly, $z(\mathbf{x}^2) = 5000$ as expected. In the same figure is shown the values of λ that will make the basis $\mathbf{J}_1$ optimal and the range that will make $\mathbf{J}_2$ optimal, as defined by the nonnegativity requirements outlined above. We call $\Lambda(\mathbf{J}_1)$ and $\Lambda(\mathbf{J}_2)$ the weight-cones. This two-dimensional case is quite graphic, introducing the concepts that can easily be carried over to higher dimensional cases.

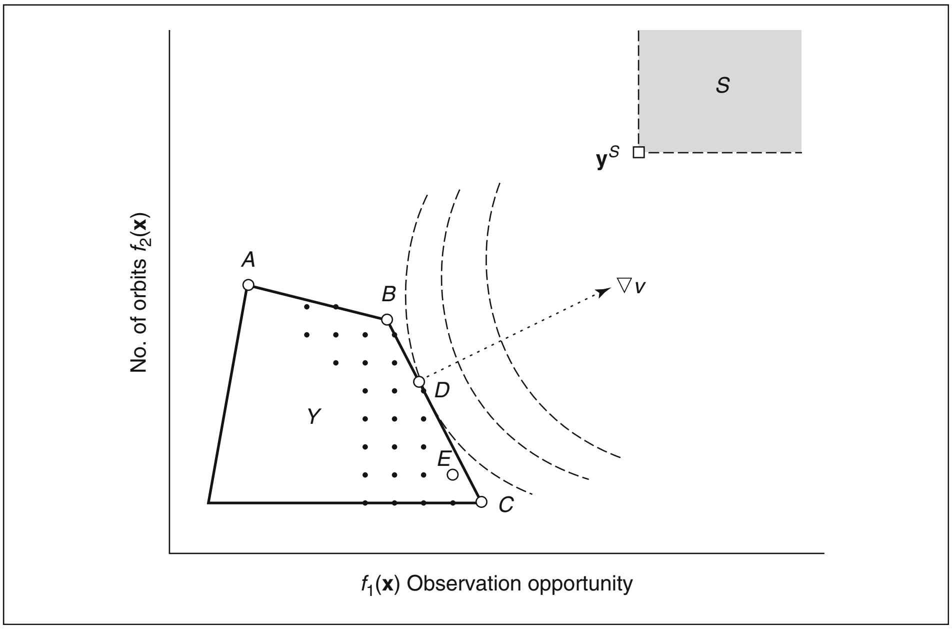
The numerical and graphical explanation of the solution procedure has one advantage. It shows quite clearly that such a multiplex algorithm for multicriteria LP is nothing but an extension of the single objective simplex algorithm. Theoretically, one can solve multiplex models that are just as large as the largest single objective models that may be solved (Ignizio and Cavalier 1994). Obviously, the computation associated with higher dimensional cases is best performed by specialized computer software. To date, such software is still in the developmental stage. An example software is distributed under the name of ADBASE (Steuer 1986), which has been extended by Shields and Chan (1991). Interactive visualization programs have also been implemented, such as one by Korhonen and Laakso (1986), marketed under the trade name VIG, which solves nonlinear programming problems in addition to LP. Such interactive techniques represent an exciting area of research for these types of multicriteria optimization problems (Buchanan and Daellenbach 1987).

B. Nonlinear and Integer Programming

The ideas introduced in MCLP can be carried over to nonlinear programming (NLP) and integer programming (IP) as well. Consider the illustration in Figure 5.8, where a nonlinear multicriteria optimization problem is shown. Although somewhat transparent from the figure, it needs to be reemphasized that not all convex combinations of the extreme points in the efficient solutions N_{ex} = {A, B, C} can be non-dominated, and that the optimal solution can be any *N*-point, not just the extreme points N_{ex}. It is not necessarily an N_{ex}-point. An example of the latter case is shown in point *D* in the nonlinear programming illustration in Figure 5.8. These two facts are further driven home by an integer program, where discrete points, rather than the feasible *Y′* region as shown, are the candidates for the optimum. In the case of IP, the efficient points may be unsupported in that they are not on the efficient frontier *A-B-C* in general. Another complication is that to locate these efficient *N*-points, the constraint reduced feasible region method as described in the introductory section on MCLP is mandatory. The weighted-sum approach, which combines the criterion functions into one, will end up missing *N*-points on the frontier, as illustrated by the point *E*.

At *D*, the value function *v* represents the tradeoff of criterion y'_1 with another y'_2. The correct weights would cause the composite-*z* gradient $\nabla_v^T = \left(\dfrac{\partial v}{\partial y'_1}, \dfrac{\partial v}{\partial y'_2}\right)$

Figure 5.8 A MORE GENERAL MULTICRITERIA OPTIMIZATION PROBLEM


S
$\mathbf{y}^S$
A
B
∇v
D
Y
E
C
No. of orbits $f_2(\mathbf{x})$
$f_1(\mathbf{x})$ Observation opportunity

to point in a direction normal to BC. This concept is akin to linear programming (LP), in which the gradient of the objective-function z is indicated by the vector **c**, in other words, $\nabla^T z = \left(\frac{\partial z}{\partial x_1}, \frac{\partial z}{\partial x_2}, \ldots, \frac{\partial z}{\partial x_n}\right) = (c_1, c_2, \ldots, c_n)$. Take the tangent at point D: $v = \lambda'_1 y'_1 + \lambda'_2 y'_2 = constant$. The slope of the tangent bears resemblance to the relationship derived for the linear additive value function, where the gradient of the value function is perpendicular to the tangent:

$$\frac{\partial v}{\partial y'_1} = \lambda'_1, \quad \frac{\partial v}{\partial y'_2} = \lambda'_2 \tag{5.1}$$

hence the slope of the gradient is

$$\frac{\lambda_2}{\lambda_1} = \left(\frac{\partial v}{\partial y_2}\right) \Big/ \left(\frac{\partial v}{\partial y_1}\right) \tag{5.2}$$

In multicriteria nonlinear optimization, solution procedures are by and large built upon this gradient concept when the criterion functions and/or value functions are no longer linear. This concept can be further extended to the case when the constraints are nonlinear in addition (Li and Wang 1994).

In general, the state of the art in multicriteria nonlinear and integer programming is not at all as developed as MCLP, which in and of itself is already complex. In an MCLP problem of any size, the N-set is often huge, so much so that wading through the set is no trivial task (Karasakal and Köksalan 2009). Imagine now that we proceed to the integer and nonlinear cases, which further introduce complexity of their own to the problem (Karaivanova et al. 1992; White 1990). However, real world problems of facility location, however, often belong to the integer and nonlinear cases. We will illustrate each of these subsequently.

C. An Interactive Frank-Wolfe Example

A prominent way of performing **multicriteria nonlinear programming** is the interactive Frank-Wolfe approach. Building upon the piecewise linear concept outlined in Chapter 4, this approach assumes the existence of an underlying preference function, but never actually requires this preference function to be identified explicitly (Hokkanen et al. 1999). The basic idea is that even if the decision maker cannot specify an overall preference function, he or she can provide local information regarding a preference at a particular situation. The iterative approach moves from an initial feasible solution toward optimal solution by finding the direction of steepest ascent and the optimal step size in that direction. Again, explicit knowledge of the overall preference function is not essential. Only local information concerning the preference of the decision maker is required, and this, in turn, is sufficient to determine the direction and step size.

Refer to the airport location problem discussed in Chapter 4. The airport location problem calls for the selection of a site between the two cities of Cincinnati and Dayton, Ohio—with populations of two and one million respectively—so that both travel time and noise impact are minimized. To apply the F-W method to this problem, two criterion functions, travel cost $f_1(\mathbf{x})$ and noise impact $f_2(\mathbf{x})$, are known. The following condition must exist: all $f_i(\mathbf{x})$, $i = 1, 2, \ldots, q$, are convex and continuously differentiable in their respective domains, and the constraints form a convex and **compact set** (i.e., a contained region to prevent unboundedness). Here, $q = 2$, and $f_1(\mathbf{x})$ is the travel time and $f_2(\mathbf{x})$ is the noise. By formulating the airport location problem as minimizing travel and noise, we rewrite the two-city case as

$$\begin{array}{ll} \text{Min } v(f_1(\mathbf{x}), f_2(\mathbf{x})) = v(2x_1 + x_2, 2x_1^{-2} + x_2^{-2}) & \\ \text{s.t.} \qquad\qquad\qquad\qquad x_1 + x_2 \geq 60 & \\ x_i \geq 0 \quad (i = 1, 2) & \end{array}$$

where x_1 is the distance from Cincinnati and x_2 is the distance from Dayton.

Taking the gradient of value function $v(\mathbf{f}) = \lambda'_1 f_1(\mathbf{x}) + \lambda'_2 f_2(\mathbf{x})$ yields

$$\nabla_{\mathbf{x}} v(\mathbf{f}) = \frac{\partial v}{\partial f_1}\begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} \\ \dfrac{\partial f_1}{\partial x_2} \end{bmatrix} + \frac{\partial v}{\partial f_2}\begin{bmatrix} \dfrac{\partial f_2}{\partial x_1} \\ \dfrac{\partial f_2}{\partial x_2} \end{bmatrix} \tag{5.3}$$

which is evaluated at the sequence of locations $\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2, \ldots$ and so forth. The initial gradient at the halfway point between Cincinnati and Dayton $\mathbf{x}^0 = (30, 30)^T$, for example, is

$$\nabla_{\mathbf{x}} v(f(\mathbf{x}^0)) = \lambda'_1 \begin{bmatrix} 2 \\ 1 \end{bmatrix} + \lambda'_2 \begin{bmatrix} -4x_1^{-3} \\ -2x_2^{-3} \end{bmatrix}_{\mathbf{x}^0 = (30, 30)} = \begin{bmatrix} 2\lambda'_1 - 0.00015\lambda'_2 \\ 1\lambda'_1 - 0.00007\lambda'_2 \end{bmatrix} \tag{5.4}$$

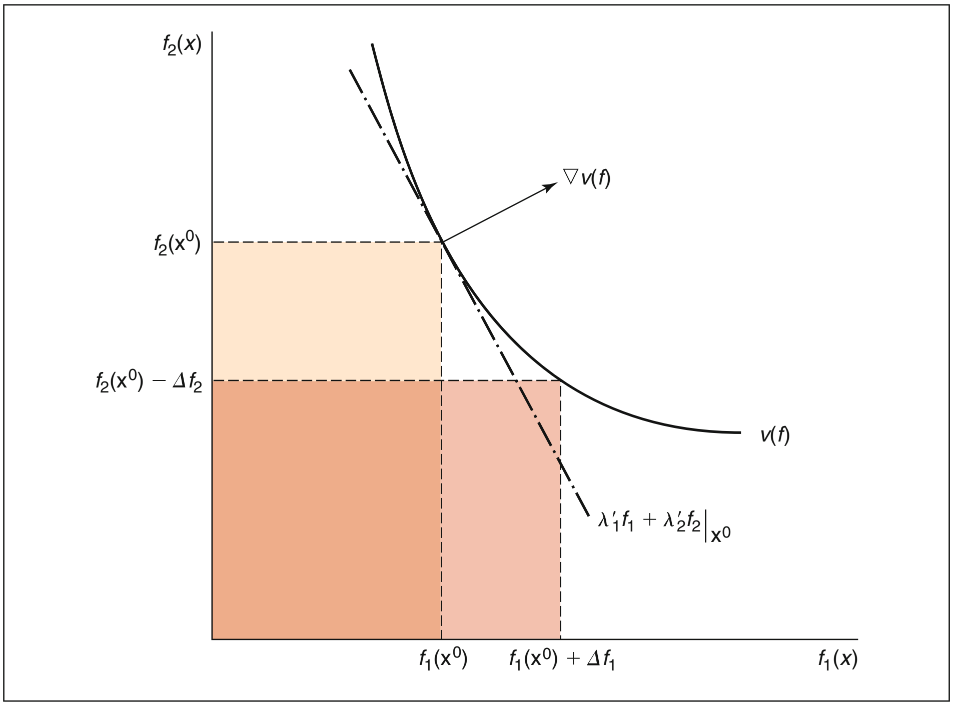
and the LP to be solved is simply

$$\underset{\mathbf{x} \in X}{\text{Min}}\ \nabla_x^T v\ (f(\mathbf{x}^0))\ \mathbf{x} = \underset{\mathbf{x} \in X}{\text{Min}}\ (2\lambda'_1 - 0.00015\lambda'_2, \lambda'_1 - 0.00007\lambda'_2)\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{5.5}$$

where $\mathbf{x} \in X$ is a shorthand notation for $x_1 + x_2 \geq 60$ and $x_i \geq 0$. Suppose the decision maker decides that the **marginal rate of substitution** is fifty-fifty, or $\lambda'/\lambda' = -1$, through local linearized indifference curves such as the one shown in Figure 5.9. In this figure, the decision maker is asked about the increment of travel cost $\Delta f_1(\mathbf{x})$ for which he or she is willing to trade against a decrement of aircraft noise $\Delta f_2(\mathbf{x})$. The slope of this indifference curve is precisely $-\lambda'_1/\lambda'_2$, which in the case of equal weights assumes the value of -1. Without loss of generality, let us set $\lambda'_1 = 1$, which means $\lambda'_2 = 1$ in this example. (Here $\lambda'_1 + \lambda'_2 \neq 1$.) Now by the following LP, the optimal solution $\mathbf{x}^* = (0, 60)^T$ is determined: Min $\{2x_1 + x_2 \mid x_1 + x_2 \geq 60; x_i \geq 0\}$. Thus at this initial iteration of the algorithm we are moving the airport toward Cincinnati from the halfway point between the two cities according to the steepest ascent direction $\mathbf{d}^0 = \mathbf{x}^* - \mathbf{x}^0$, where $\mathbf{x}^0 = (30, 30)^T$ and $\mathbf{x}^* = (0, 60)^T$ or $\mathbf{d}^0 = (-30, 30)^T$.

The decision maker now determines the step size α to move along this direction $\mathbf{x}^0 + \alpha^0\mathbf{d}^0$. The decision maker, assisted by tabular or graphic displays of the function $f(\mathbf{x}^0 + \alpha^0\mathbf{d}^0) = (f_1(\mathbf{x}^0 + \alpha^0\mathbf{d}^0), f_2(\mathbf{x}^0 + \alpha^0\mathbf{d}^0))$, determines the step size α^0 between 0 and 1. One possible way to obtain the best step size α is to display the

Figure 5.9 DETERMINATION OF MARGINAL RATE OF SUBSTITUTION



values for the two criterion functions $f_i(\mathbf{x}^0 + \alpha^0\mathbf{d}^0)$ for $i = 1$ and 2 as a function of α over the selected values of α in a tabular or graphic way. Example of the curves are shown in Figure 5.10. For example, the graph for travel is a linear function of α, as shown by the equation $f_1(\mathbf{x}^0 + \alpha\mathbf{d}^0) = 2(30 + [-30]) + (30 + \alpha[30]) = 90 - 30\alpha$. The decision maker then determines a value of α for the most preferred values of the corresponding criterion functions. In short, the following optimization problem is solved: $\text{Min}_{0 \le \alpha \le 1}\ v(\mathbf{f}(\mathbf{x}^0 + \alpha\mathbf{d}^0))$. Let the stakeholder(s) read off the $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ values on Figure 5.10. From these f_is, the λ'_1 and λ'_2 values can be determined in

Figure 5.10 GRAPHIC DISPLAY TO DETERMINE STEP SIZE

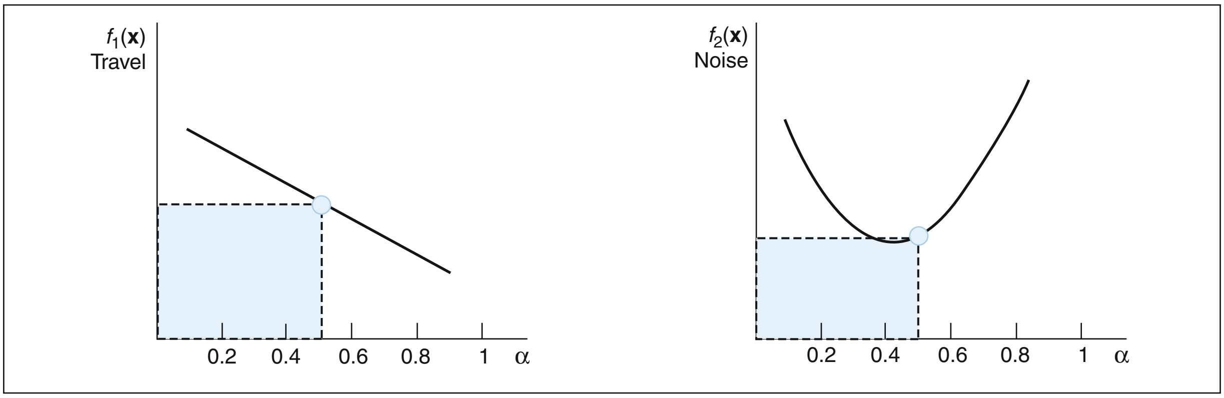


Figure 5.9 by reading off the slope λ_1'/λ_2' at $(f_1(\mathbf{x}_0), f_1(\mathbf{x}_0))$. The λs facilitate the next iteration. Suppose $\lambda^0 = 0.5$. We are now at $\mathbf{x}^1 = \mathbf{x}^0 + \alpha^0\mathbf{d}^0 = (30, 30)^T + 0.5(-30, 30)^T = (15, 45)^T$ and the iterations continue until the incremental ascent of the preference function v is minuscule, as with most hill-climbing algorithms as explained in Chapter 4.

Assuming equal weights among the two criterion functions and a constant step size of 0.5, a series of iterations were performed, with the following results (Staats and Chan 1994):

Iteration k	airport location $\mathbf{x}^k$
0	$(30, 30)^T$
1	$(15, 45)^T$
2	$(7.5, 52.5)^T$
3	$(3.75, 56.25)^T$
4	$(1.875, 58.125)^T$
5	$(0.9375, 59.0625)^T$
6	$(30.46875, 29.53125)^T$
.	.
.	.

It can be seen that while the airport steadily moves toward Cincinnati in the first five iterations, starting at iteration 6, there is a direction reversal toward Dayton. If we make subsequent iterations, it will begin moving toward Cincinnati again. However, there is a limit to this westward movement—namely at a point east of the previous reversal point $(0.9375, 59.0625)^T$. This point is further from Cincinnati and closer to Dayton, or $x_1 > 0.9375$ and $x_2 < 59.0625$. As this point subsequently moves east toward Dayton again, there is once again a limit to its movement. In this case, it falls short of (30.46875, 29.53125). Thus the airport location bounces back and forth within a shrinking interval, eventually converging toward a final equilibrium point. This point can be determined by a shortcut method for this case of equal weights ($\lambda_1' = \lambda_2' = 1$). This is the point where the objective function of the LP as shown in Equation 5.5 is minimized for both x_1 and x_2, which occurs when the gradient $\nabla_{\mathbf{x}} v$ is $(1, -1)^T$, or when the coefficient for x_1 is the same as x_2 in the objective function: $2\lambda_1' + \lambda_2'(-4/x_1{}^2) = \lambda_1' + \lambda_2'(-2/x_2{}^3)$. Thus for $\lambda_1' = \lambda_2' = 1$, we solve the equation set consisting of $2 - 4/x_1^3 = 1 - 2/x_2^3$ and $x_1 + x_2 = 60$, where $x_1 + x_2 \geq 60$ has to be satisfied at strict equality. This equation set yields $x_1 = 1.5873$ and $x_2 = 58.4127$. This says that for equal weights placed on the two criteria, the decision maker is indifferent about noise and travel at an airport located at (1.5873, 58.4127), or about 1.6 miles (2.56 km) outside Cincinnati.

It can be shown that the above iterative procedure is a special case of a more unified interactive multiple objective programming procedure (Gardiner and Steuer 1993). This airport example has been solved previously in Chapter 4. The solution obtained here is consistent with and close to the previous solution. Because of the peculiarity of the criterion functions $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$, the gradient of $f_2(\mathbf{x})$, $\begin{bmatrix} -0.00015 \\ -0.00007 \end{bmatrix}$, is small for the $\mathbf{x}^0$ chosen in comparison with $f_1(\mathbf{x})$, $\begin{bmatrix} 2 \\ 1 \end{bmatrix}$. For this reason, the example does not fully illustrate the importance of properly determining λ_1' vis-a-vis λ_2' well. Perhaps a better example is to solve the equivalent problem

$$\begin{aligned} \text{Max } & v(f_1(\mathbf{x}), f_2(\mathbf{x})) = v(-2x_1 - x_2, 2x_1^2 + x_2^2) \\ \text{s.t.} & \quad x_1 + x_2 = 60 \\ & x_i \geq 0 \quad (i = 1, 2) \end{aligned}$$

where $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ are concave and a maximization objective replaces a minimization. This will illustrate the interactive F-W procedure better, when the decision maker participates—in a more significant way—in determining the direction of climb and the step size. Most interestingly, the solution to this new problem is diametrically opposite to the previous minimization formulation, given the same λ' and α values, and given that the noise function $f_2(\mathbf{x})$ so formulated is different from the one used in the previous $f_2(\mathbf{x})$. In this case, the airport is located at $(59.50,0.5)^T$ or half a mile (0.8 km) outside Dayton.

Readers interested in more detailed discussions of interactive multi-objective programming can consult Seo and Sakawa (1988) and Gardiner and Steuer (1993). It is apparent from the above example that interactive programming of this sort is highly numerical in nature. Aside from convergence issues, different functional forms may give rise to drastically different solutions, as one can see clearly from the airport-location problem above. In practice, however, such an extreme result is unlikely to occur. Remember that the above results are built upon the rather indefensible assumption of $\alpha = 0.5$ and $\lambda'_1 = \lambda'_2 = 1$ throughout the iterations. This simplifying assumption is made mainly for our computational convenience. In fact, the entire foundation of interactive procedures of this sort is to explore the decision maker's revealed preferences, guided by charts and tables, at various situations. Correspondingly, his or her reactions, as reflected by values of α and λ's—are expected to be different at each iteration. Convergence in this case is obtained not so much from numerical properties as it is from the decision-maker's behavioral changes. The behavioral change from one iteration to another, reflecting sharpening of the decision maker's focus, should more than compensate for the dilemma that apparently arose from different functional forms for the travel and noise criterion functions $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$, leading toward a consistent location for the common airport between Cincinnati and Dayton.

D. Comments

The most challenging (and interesting) part of MCDM is still the Z'-space, where the criteria are to be traded against one another. Generally, there have to be at least two criteria for decision making to occur, since a single criterion means simply "take it from the top" on a uni-dimensional scale–a laborious exercise at best. In spite of the seemingly elaborate effort made above, let us conclude this section by reiterating an important point. It is necessary to have good measurement units for the metric used. Advanced computational algorithms are also necessary for efficient search, but they are not sufficient for decision making in the presence of multiple criteria. The crux of MCDM lies in the Z'-space, making participatory, interactive techniques so much more attractive as a solution tool.

We have illustrated the interactive procedure for NLP above using the F-W method. A body of knowledge exists for evaluating a finite set of discrete alternatives. One such procedure is ELECTRE (Roy 1977), which is a robust technique that does not necessarily assume transitivity of preferences. Of the variants to the method, Chankong and Haimes (1983) recommended the sequential (interactive) elimination procedure inasmuch as it furnishes opportunities to gain greater understanding and appreciation of what is being done, and more importantly, what levels of risk are involved when eliminating certain alternatives. Chan (2005) discusses this method in the "Facility Location" chapter, but he uses a deterministic, outranking elimination procedure.

V. GOAL SETTING

The type of problems we have been solving above are often referred to as **goal seeking,** where "the more the merrier" is the modus operandi. **Goal setting,** on the other hand, refers to an environment in which there is a standard against which alternatives can be compared. For example, in locating a satellite tracking station, there are minimum standards one sets for observational opportunity and coverage of the various orbits. A station location either satisfies this minimum standard or it does not. Thus goal setting is defined as the procedure of identifying a satisficing set *S* such that, whenever the decision outcome is an element of *S*, the decision maker will be happy and satisfied and is assumed to have reached the optimal solution.

A. Compromise Programming

We now refer to the example in Figure 5.8 again. Assuming the decision maker defines his satisficing set by $S = \{(y'_1, y'_2) \mid y'_1 \geq 20, y'_2 \geq 10\}$, meaning that the minimal standard for observational opportunity is 20 (in say a maximum of 100) and the station needs to track at least 10 orbits. The graphical depiction clearly shows that no satisficing solutions exist, since the region Y' and S do not intersect. The logical solution is to look for the second best, similar to the examples shown in Figures 5.1, 5.2 and 5.3, where deviation from the threshold standard $\mathbf{y}^s = (20, 10)^T$ is to be minimized here. The goal-setting (GS) program now looks like

$$\begin{gathered} v = \text{Min } (d_1 + d_2) \\ f_1(\mathbf{x}) + d_1 \geq 20 \\ f_2(\mathbf{x}) + d_2 \geq 10 \end{gathered}$$

with the feasible region $\mathbf{x} \in X$ as defined previously for this example. Notice here v assumes a particular goal setting value function, in which the two criteria, $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ (and hence the deviational variables, d_1 and d_2) are in different units. The total displacement from *S*, defined here as the simple sum (rather than say the weighted sum) of the two deviations, is minimized (Lai et al. 1994).

There is no station locations that can satisfy the minimal standard, or $S \cap Y' = \varnothing$ and $v > 0$. To find a satisficing solution, we must restructure the feasible region X in the decision space, change the criterion vector $\mathbf{f}$ and/or alter the standards that define *S*. In other words, we must either examine more locations for the tracking station, or improve the tracking capabilities for an existing site, or our expectation for observational opportunities and orbit coverage must be lowered.

If X and **f** are fairly fixed, we may need to change S. For example, the goals of $y'_1 \geq 20$ and $y'_2 \geq 10$ must come down. This is another example of an interactive process between the decision maker and the analyst.

Thus far, we have ignored the precise way used to measure deviation. We simply stated that it is the sum of two different scales, one measured in the horizontal and the other in the vertical dimension of the outcome space. This **Manhattan metric** is distinctly different from the Euclidian metric used in Figures 5.1, 5.2, and 5.3 (Saber and Ravindran 1996). The natural question then is: what is the proper measure?

B. Deviational Measures

Different problems dictate the use of different deviational measures, since the way the compromise is measured defines the value function. It so turns out that both of these deviation metrics—Manhattan and Euclidean—can be accommodated within the l_p-metric, which also includes the weights placed on y'_1 and y'_2. The metric assumes the functional form of

$$r'(\mathbf{y}'; p, \mathbf{w}') = \left\| \mathbf{y}' - \mathbf{y}'^{*} \right\|_{p,\mathbf{w}'} = \left[\Sigma_i w'^{p}_{i} \left| \mathbf{y}'_1 - \mathbf{y}'_i \right|^{p} \right]^{1/p} = \left[\Sigma_i w'^{p}_{i} d^{p}_{i} \right]^{1/p} (1 \leq p \leq \infty)$$

In the above expression, it can be seen that it reduces to the Manhattan and Euclidean metrics when $p = 1$ and 2 respectively. In the case of l_∞-metric, the above expression is simply $\text{Max}_i \left[w'_i d_1 \right]$ When $\mathbf{w}'$ is not specified, the usual convention is to assume $w_i' = 1$ for all i's.

In general, as p is increased from 1 to 2, the emphasis shifts toward the more prominent of the Ith $\mathbf{y}'$ component. In the extreme case when $p = \infty$, only the more prominent component counts, with the less prominent, albeit just a shade less prominent, totally overwhelmed. We refer to this case as the totally noncompensatory situation. Using the example above about satellite tracking stations, it may turn out that $f_2(\mathbf{x})$, the number of orbits covered, may be the criterion that the decision maker really cares about whenever there is a site that has an $f_2(\mathbf{x})$ advantage over $f_1(\mathbf{x})$ (the number of observational opportunities). This phenomenon should then be modeled as a compromise program with an l_∞-metric.

In location models, the l_∞-metric is of particular interest. If y'_1 and y'_2 epresent distances demands 1 and 2 are from a facility, it can be shown that it can be modeled as a special case of a compromise program with the l_∞-metric. In this case $r'(\mathbf{y}'; \infty, \mathbf{1})$ boils down to the further of the two distances. In emergency facility location, such as the siting of a fire station, for example, it is common to minimize the furthest distance away, so that the worst situation can be covered—in case a fire breaks out at the furthest house from the station. Again, such a way of fighting fire reflects a philosophical viewpoint of caring for the most geographically disadvantaged household in the community. An equally valid figure of merit may be to minimize the average response time to all the households in the area. If this is the case, the Manhattan metric is definitely a viable measure in a city with a square grid street system. It can further be argued that the weights should not be equal among all parts of town, that highly populated areas should receive more attention than the wilderness. Thus the weights w come in besides the parameter p. The geometric interpretation of l_∞-metric is found in Chan (2005) under the "Facility Location" chapter.

C. Goal-Setting Example

To close out the discussion on goal setting, a numerical example may be in order. Consider the compromise program:

$$\begin{array}{ll} \text{Goal 1:} & f_1(\mathbf{x}) = x_1 \geq 8 \\ \text{Goal 2:} & f_2(\mathbf{x}) = x_2 \geq 9 \\ \text{s.t.} & 3x_1 + x_2 \leq 24 \\ & 2x_1 + 7x_2 \leq 35 \\ \text{all } x\text{s} & \text{positive.} \end{array}$$

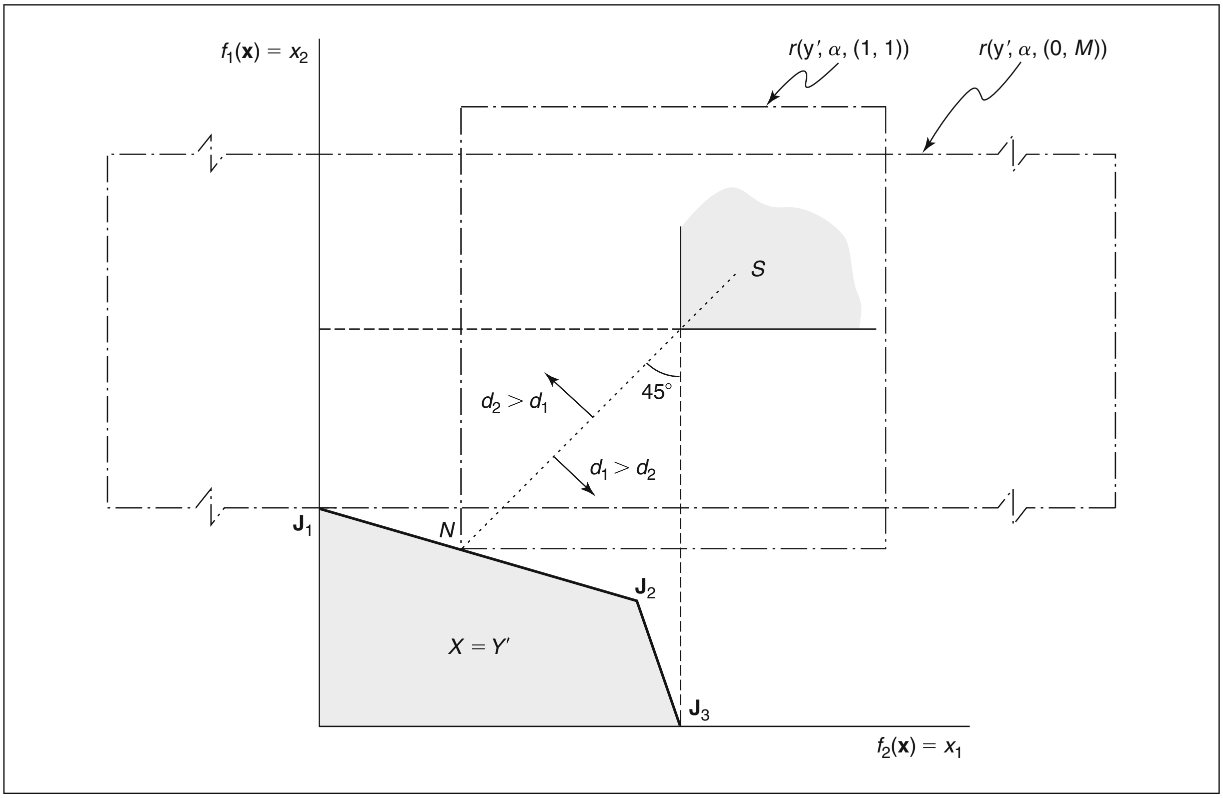
(a) Graph the X space, Y′ space and the satisficing set *S*.

The identical *X* and *Y*′ spaces are sketched in Figure 5.11, complete with the satisficing set S and the extreme points of *X* and *Y*: $\mathbf{J}_1 = (0, 5)$, $\mathbf{J}_2 = (7, 3)$, and $\mathbf{J}_3 = (8, 0)^T$.

(b) Assuming both goals at the same priority level, specify the point in both *X* and *Y*' that minimizes the maximum deviation.

This corresponds to minimizing the l_∞-metric or more specifically Min $[r'(\mathbf{y}'; \infty, \mathbf{1})$ = Max $\{(8 - y_1'), (9 - y_2')\}$ = Max $(d_1, d_2)]$. It can be verified that this amounts to point $N = (28/9, 37/9)$, which is obtained by drawing a square box centered at (8, 9) that

Figure 5.11 GOAL-SETTING EXAMPLE



barely touches ($\mathbf{J}_1$, $\mathbf{J}_2$). The square box is the contour of the l_∞-norm, whose horizontal edge is the locus of $d_1 > d_2$ and the vertical $d_2 > d_1$. The watershed is the 45^0 diagonal.

(c) Specify the point in both X and Y' space that solves the preemptive model with the goals ranked in the order in which they are listed. What is the solution point if we reverse the priorities?

The lexicographic ordering between the first and second goal, or $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ respectively, can be thought of as placing a very large weight M on $f_1(\mathbf{x})$ in comparison to $f_2(\mathbf{x})$. This amounts to Min [$r'(\mathbf{y}'; \infty, (M, 0) = \text{Max}\{M(8 - \mathbf{y}'_1), 0(9 - \mathbf{y}'_2)\} = \text{Max}\,(8 - y'_1)$]. This yields the non-dominated point J_3. By reversing the order, we write Min [$r'(\mathbf{y}', \infty, (0, M) = \text{Max}\{0(8 - \mathbf{y}'_1),\ \text{M}(9 - \mathbf{y}'_2)\} = \text{Max}\,(9 - \mathbf{y}'_2)$], resulting in another non-dominated point $\mathbf{J}_1$. Again, these two points can be located geometrically by drawing two boxes, with the first one greatly elongated vertically, while the second elongated horizontally, touching X or Y' at $\mathbf{J}_3$ and $\mathbf{J}_1$ respectively.

Now imagine gradually shifting the weight between the two extreme values of 0 and M. A series of boxes can be drawn between $\mathbf{J}_1$ and $\mathbf{J}_3$ that sketch out the entire efficient frontier (or N-set) of X and Y', with the intermediate point N corresponding to equal weights among f_1 and f_2. In other words, minimizing the l_∞-metric as a quasi-convex function can generate any non-dominated solution point between $\mathbf{J}_1$ and $\mathbf{J}_3$, including $\mathbf{J}_2$. Notice applying the l_∞-metric to the example shown in Figure 5.8 will similarly sketch out the N-set along *A-B-C*.

VI. VALUE FUNCTIONS

Thus far, we have been alluding to value functions through our discussion of the Z'-space and the deviational measures such as the l_p-norm. We also have been promulgating the fact that the heart of MCDM lies in dealing with the value function. We certainly have arrayed some useful tidbits about MCDM without facing the hard problem! Maybe it is time for us to face the central issue by giving a formal definition: A value function $v(\mathbf{y}')$ on Y' is a metric that alternative 1 is preferred to 2, or $\mathbf{y}^1 > \mathbf{y}^2$, if and only if $v(\mathbf{y}^1) > v(\mathbf{y}^2)$. Utility function is a special case when uncertainty in the outcomes y_i's is involved, through which the risk perception of a decision maker is elicited. While value functions can be used as an ordinal scale to rank order alternatives, utility function is a cardinal scale to compare the merit of one alternative to another. For the purpose of this book, we use the generic term value function to include both ordinal and cardinal measurements. More will be said about this later.

A. Additive versus Multiplicative Form

A value function represents revealed preference information. A multi-attribute value function $v(y'_1, \ldots, y'_q)$ assumes some kind of attribute independence among y_i's. If the random variables y_i's ($i = 1, \ldots, q$) are statistically independent, the joint density function

$$P(y'_1, \ldots, y'_n) = P(y'_1) \ldots P(y'_q) \tag{5.6}$$

where $P(y_i)$ is the marginal (univariate) density function of y'_{i}. Hence the assessment of a q-dimensional function is simplified to that of q one-dimensional functions: $v(y'_1, \ldots, y'_q) = g[v_1(y'_1), \ldots, v_q(y'_q)]$. Notice the cross terms $v(y'_1, y'_2)$, $v(y'_1, y'_3)$, and so forth, are absent in the $g(.)$ function. Thus the independence property greatly simplifies the determination of multi-attribute value functions. As will be shown, additive value functions look like $w_1v_1(y'_1) + w_2v_2(y'_2) + w_3v_3(y'_3)$, while multiplicative value functions look like $w_1v_1(y'_1) + w_2v_2(y'_2) + w_3v_3(y'_3) + kw_1w_2v_1(y'_1)v_2(y'_2) + kw_1w_3v_1(y'_1)v_3(y'_3) + kw_2w_3v_2(y'_2)v_3(y'_3) + k^2w_1w_2w_3v_1(y'_1)v_2(y'_2)v_3(y'_3)$, where ws are generalized weights, and ks are scaling constants (Sainfort and Deichtman 1996). In previous examples of the value function, we have assumed $v_i(y'_i) = y'_i$, which greatly simplifies the expressions for both additive and multiplicative value functions.

Value functions v_is are scaled from 0 to 1, and the role of k in a multiplicative value function is to assure that this compound value function v will also assume value in the interval 0 to 1. If $w_i = \lambda'_i$, or $\Sigma_i w_i = 1$, then $k = 0$, and the multiplicative form reduces to the additive form. Only when $w_i \neq \lambda'_i$ does $k \neq 0$, hence the need for a multiplicative value function. Notice w_i is nonnegative and k can be negative in value. While there are a whole host of equations to represent a value function, the advantage of the additive and multiplicative forms is that most conceivable shapes of the function can be accommodated within the order of q calibration constants ws and ks.

Again, we like to emphasize that the ranking among alternatives will be the same whether an additive or multiplicative function is used, as long as they are strategically equivalent. The units of a value function—to be differentiated now from a utility function—have no intrinsic meaning inasmuch as we are dealing with ordinal ranking. Any value function can be transformed by a monotonic function, and the result will represent exactly the same preferences as before. An example is $v(y'_1, y'_2) = y'_1y'_2$. Taking a logarithm of the value function will yield another value function $v'(y'_1, y'_2) = (\ln y'_1) + (\ln y'_2)$. Both v and v' will give the same preference ranking among alternatives since a logarithmic function is a monotonic transformation. The time we worry about the exact functional form is when cardinal measurements, sometimes referred to as the preference intensity, are required. It also follows from this discussion that the establishment of a value function implies transitivity of preference. Value functions, where only ordinality is involved, are not measured directly. This is a consequence of the observation that quite different functions may be strategically equivalent and that units of value have no intrinsic meaning as mentioned. For the purpose of evaluation, the information contained in a value function can be obtained indirectly, and this is done by estimating the decision maker's revealed preferences at sampled situations. An example is in the interactive Frank-Wolfe method discussed earlier, where the exact form of the value function is not known, only attribute and criterion tradeoffs at local situations are assessed.

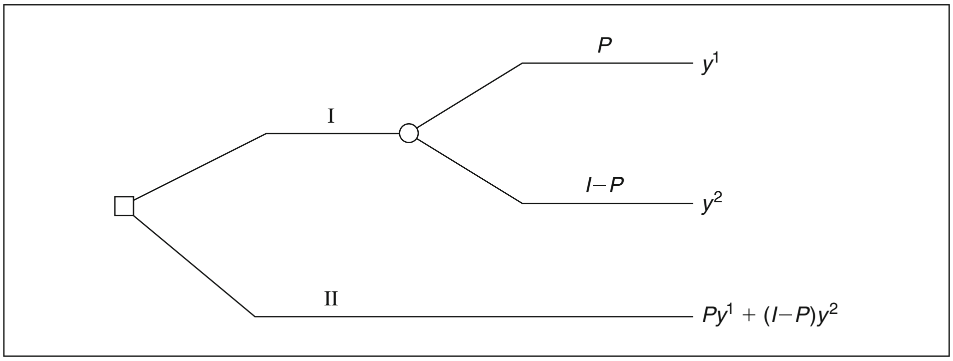
B. Univariate Utility Function Construction

Measurement of utility for an alternative is based on the axiom

$$E(v(\mathbf{y}')) = \Sigma_j P^j v(\mathbf{y}^1) \tag{5.7}$$

This says that the utility of an alternative is the sum of the utility of each of the possible outcomes $\mathbf{y}^j$ weighted by the probability of occurrence P^j. Consider a

Figure 5.12 DECISION TREE FOR UNIVARIATE UTILITY FUNCTION



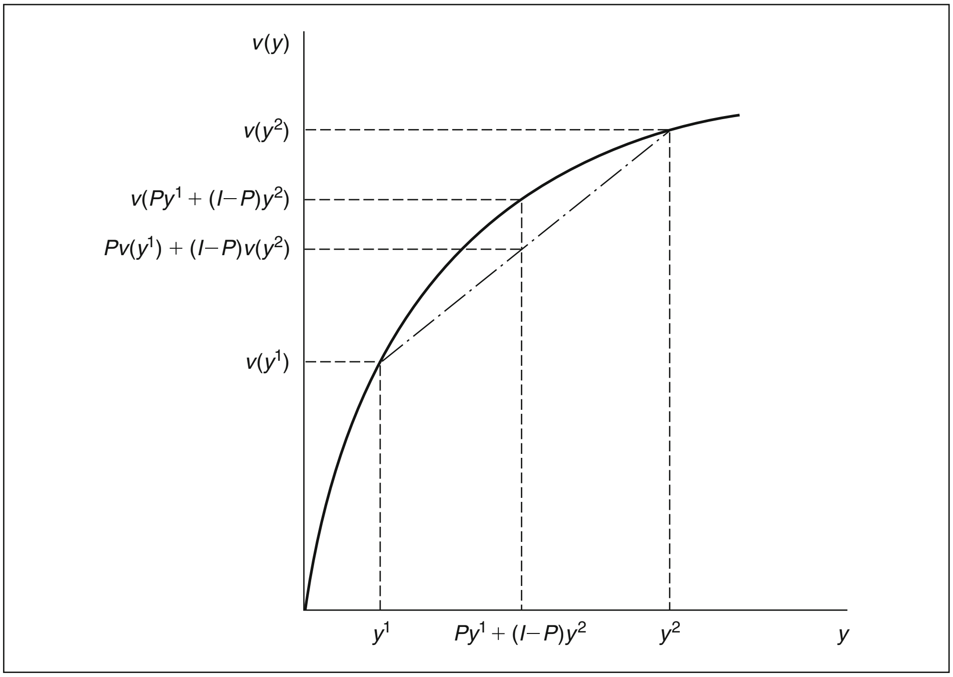
decision maker (DM) playing a lottery. The DM's choices in this lottery can be illustrated in a decision tree, which is shown in Figure 5.12. If the probability of losing the lottery is P, winning is $1 - P$, the amount he or she loses in the lottery is y^1, and the amount he or she wins is y^2, the decision maker has either option II of taking the lump sum $Py^1 + (1 - P)y^2$, or option I of playing the lottery, with an expected return of $Pv(y^1) + (1 - P)v(y^2)$.

The classic illustration of this concept is to consider the situation when there is only a single attribute $\mathbf{y}_i'$ for three probabilistic scenarios: risk-averse, risk-prone and risk-neutral. To measure a multi-attribute value function, whether additive or multiplicative, involves defining these unidimensional or univariate utility functions $v(\mathbf{y}_i')$ as a first step. The independence property among y_is facilitates straightforward aggregation of these univariate utility functions into the multi-attribute form. To simplify the notation, we write y in lieu of y_i' in the discussions here.

1. Risk-aversion example. To start out, we illustrate construction of a univariate utility function for a DM who is risk-averse. Being a conservative, the DM prefers the expected monetary value of a nondegenerate[3] lottery y(II) to the lottery itself y(I). In other words, he or she is not willing to take a chance and prefers option II to I or $y(\text{II}) > y(\text{I})$, which when translated into utility, means $v[y(\text{II})] > v[y(\text{I})]$: $v[Py^1 + (1 - P)y^2] > Pv(y^1) + (1 - P)v(y^2)$. This results in a strictly concave utility function as illustrated in Figure 5.13. Here, the utility of the expected sum of money $v(Py^1 + (1 - P)y^2)$ is greater than the expected value of the utility of winning and losing, $Pv(y^1) + (1 - P)v(y^2)$.

2. Risk-prone example. Conversely, suppose the DM prefers the lottery y(I) to the expected monetary value of a nondegenerate lottery y(II), or $y(\text{I}) > y(\text{II})$, $v[y(\text{I})] > v[y(\text{II})]$. This results in $Pv(y^1) + (1 - P)v(y^2) > v[Py^1 + (1 - P)y^2]$. The optimistic DM is then characterized by a strictly convex utility function, as shown in Figure 5.14. It follows without saying that a risk-neutral DM will have a utility function that is simply a straight line. Notice that all these univariate functions can be represented by the function $v(y) = a + be^{-cy}$, where a, b, and c are calibration constants, and c represents the degree of risk aversion. For $v(0) = 0$ and $v(1) = 1$, $v(y) = (1 - e^{-cy})/(1 - e^{-c})$. As the positive parameter c increases, the utility function becomes more convex, indicating higher risk aversion.

Figure 5.13 A STRICTLY CONCAVE UTILITY FUNCTION



3. Certainty equivalent. A certainty equivalent of a lottery is an amount such that the DM is indifferent between the lottery and the amount $\hat{y}$ for certain. Therefore, $\hat{y}$ is defined by $v(\hat{y}) = E[v(\overline{y})]$, or $\hat{y} = v^{-1}\{E[v(\overline{y})]\}$, where $\overline{y}$ is the uncertain outcome of a lottery. Notice that the certainty equivalent is not the same as the expected return $Py^1 + (1 - P)y^2$ except for a risk-neutral DM. By way of notation, it is common to write the certainty equivalent in terms of the loss and win in a lottery, y^1 and y^2 respectively: $P(y^1) \otimes (1 - P)(y^2) \sim \hat{y}$. The certainty equivalent of the convex utility function is overlaid in Figure 5.14, so is the 0–1 normalization for v common among utility functions.

Suppose we set $v(y^1) = 0$ and $v(y^2) = 1$, where $y^1 = y^{\text{Min}} = 0$ and $y^2 = y^{\text{Max}}$ in Figure 5.15. Keeping the same probability of win and loss, the point $\hat{y}$ is now placed between the loss and win amounts of the lottery and another certainty equivalent defined as: $P(y^1) \otimes (1 - P)(\hat{y}) \sim \hat{y}'$. Then we find yet another certainty equivalent by examining the interval between $\hat{y}$ and y^2, resulting in $\hat{y}''$. $P(\hat{y})) \otimes (1-P)(y^2) \sim \hat{y}''$. The result is $v(\hat{y}') = Pv(y^1) + (1 - P)v(\hat{y}) = (1 - P)^2$ and $v(\hat{y}'') = Pv(\hat{y}) + (1 - P)v(y^2) = P(1 - P) + (1 - P) = (1 - P)(1 + P)$. Subsequent points are obtained by substituting $\hat{y}'$ and $\hat{y}''$ in the binary lottery for $[y^1, \hat{y}]$ and $[\hat{y}, y^2]$ in turn. The process can be repeated as often as desirable or practical to sketch out the full function $v(y)$. We illustrate this process in Figure 5.15. It can be seen that the process is particularly simple for $P = 0.5$—a probability most people can associate with the common experience of coin flipping. The certainty equivalents so obtained also divide the utility range into halves, quarters, and so on. For this reason, this method is often called the fractile method. We will illustrate this method step by step later on in this chapter.

Figure 5.14 A STRICTLY CONVEX UTILITY-FUNCTION

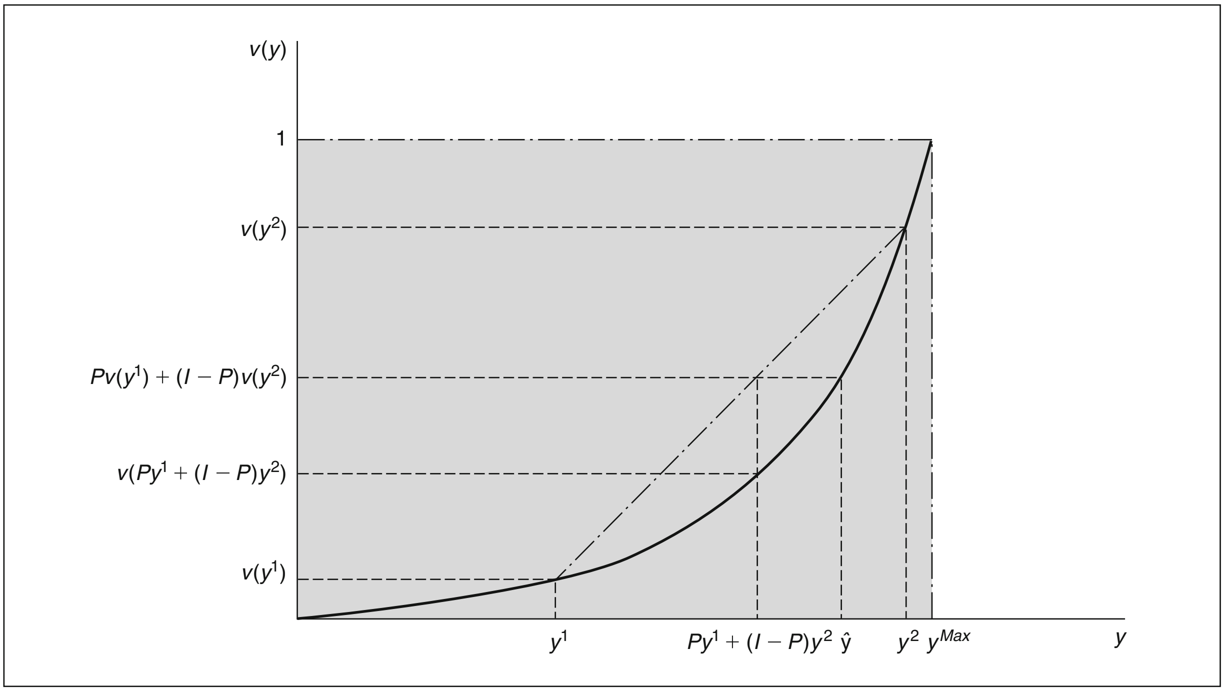
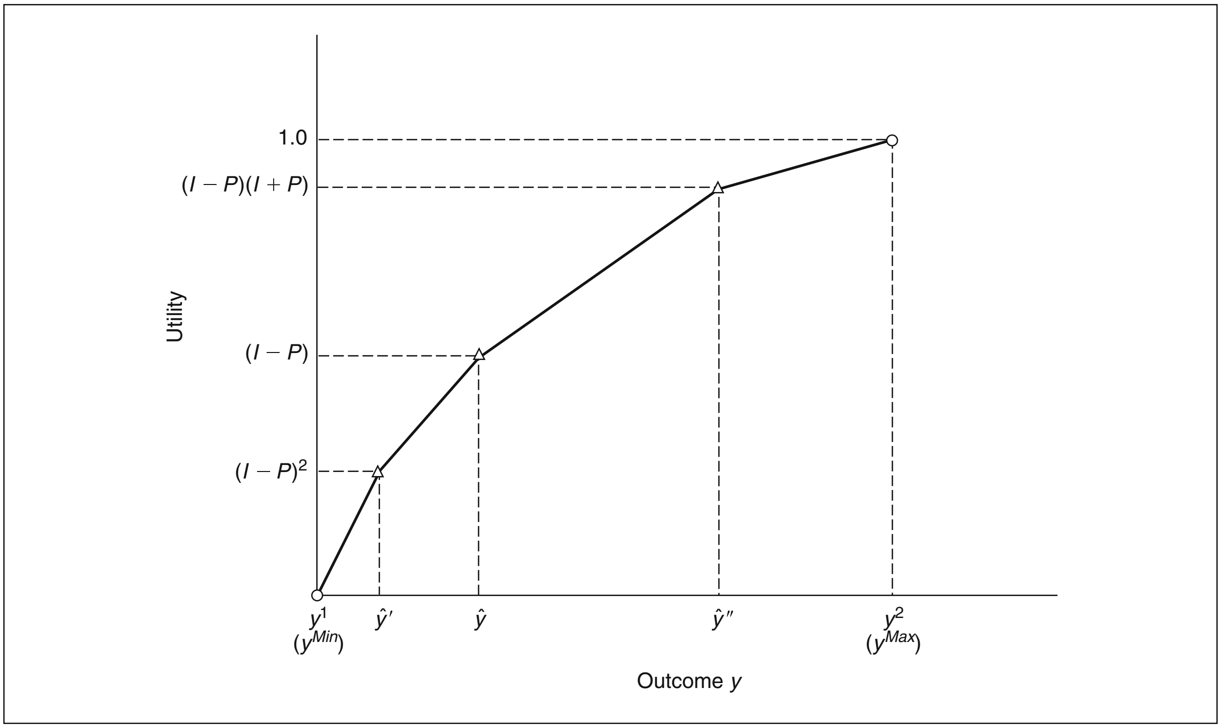


Figure 5.15 THE FRACTILE METHOD OF MEASURING UTILITY FUNCTION



C. Independence Among Criterion Functions

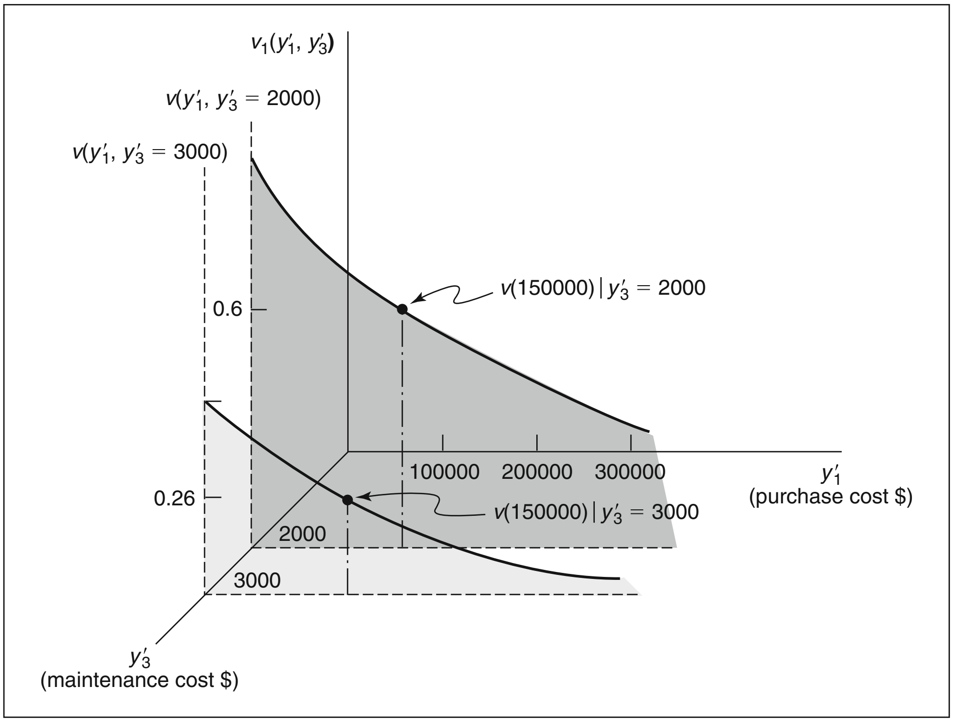
To conclude this section, let us see when a multi-attribute value function can be aggregated from a constituent set of univariate value functions constructed above, and when it is invalid to do so. Take a residential location problem. A family selects a house on the basis of purchase cost y'_1, floor space y'_2, maintenance cost (utilities and upkeep) y'_3, appreciation (%) y'_4, and appeal y'_5. The family constructs a value function for each candidate housing alternative assuming an additive value function:

$$v(\mathbf{y}') = w_1 v_1(y'_1) + w_2 v_2(y'_2) + \cdots + w_5 v_5(y'_5) \qquad (5.8)$$

where $v_i(y'_i)$s are uni-dimensional value functions.

We check the independence of attributes by asking the following mathematical question for a \$150,000 home: Is v_1(\$150,000) = constant (say 0.6) for both a maintenance cost of y'_3 = \$3000 or 2000/year? Given that both purchase cost and maintenance expenses relate to overall residential expenses, they may not be independent. The answer to the above question is "no." Equation 5.6 is violated. Hence the overall value function cannot be formed from the constituent univariate value-functions as suggested. Including both y'_1 and y'_3 in a multi-attribute value function as shown in Equation 5.8 appears not justified. (See Figure 5.16.)

Figure 5.16 TEST OF STATISTICAL INDEPENDENCE AMONG ATTRIBUTES



What happens if two criteria $f_i(\mathbf{x})$ and $f_j(\mathbf{x})$ used in an analysis are correlated? Obviously, the above two-step process of (a) measuring a univariate function and (b) aggregating univariate functions into a multivariate one will no longer be valid. More complicated analysis will have to go into forming a multi-attribute value function.

In multicriteria optimization, where we have derived the individual criterion functions $f_i(\mathbf{x})$'s, not all $f_i(\mathbf{x})$'s in the objective function $v(f_1(\mathbf{x}), \ldots, f_q(\mathbf{x}))$ can be independent. Independence in this context would mean that the $f_i(\mathbf{x})$s are orthogonal. Consider two criteria $f_i(\mathbf{x})$ and $f_j(\mathbf{x})$. A measure for the correlation between the Ith and jth criterion is similar to its statistical analogue. The angle between the two criterion vectors is defined between the gradients $\mathbf{c}^i$ and $\mathbf{c}^j$

$$\cos^{-1}\left[\frac{(\mathbf{c}^i)^T\mathbf{c}^j}{||\mathbf{c}^i||_2\,||\mathbf{c}^j||_2}\right] \tag{5.9}$$

An ideal angle, as mentioned, is 90 degrees when the two criteria are totally independent. The more the criteria are correlated, the smaller the angle.

Take the MC-simplex example from Section IV-A in this chapter, where $\mathbf{c}^1 = (5, 20)^T$ and $\mathbf{c}^2 = (23, 32)^T$ and the angle between them is computed by Equation 5.9 as 21.67 degrees. This angle agrees with the graphical plot of Figure 5.6. The angle here is too small in comparison with an ideal orthogonality of 90 degrees, indicating there is a fair amount of correlation between the two criterion-functions. For high correlations, seemingly good weighting among the criteria—λ' weights in accordance with the decision maker's priority—produce non-optimal points. On the other hand bad weights may produce an optimal point. This result is completely analogous to the statistical discussion above where the lack of independence (or spurious correlations) will result in a highly complex modeling task.

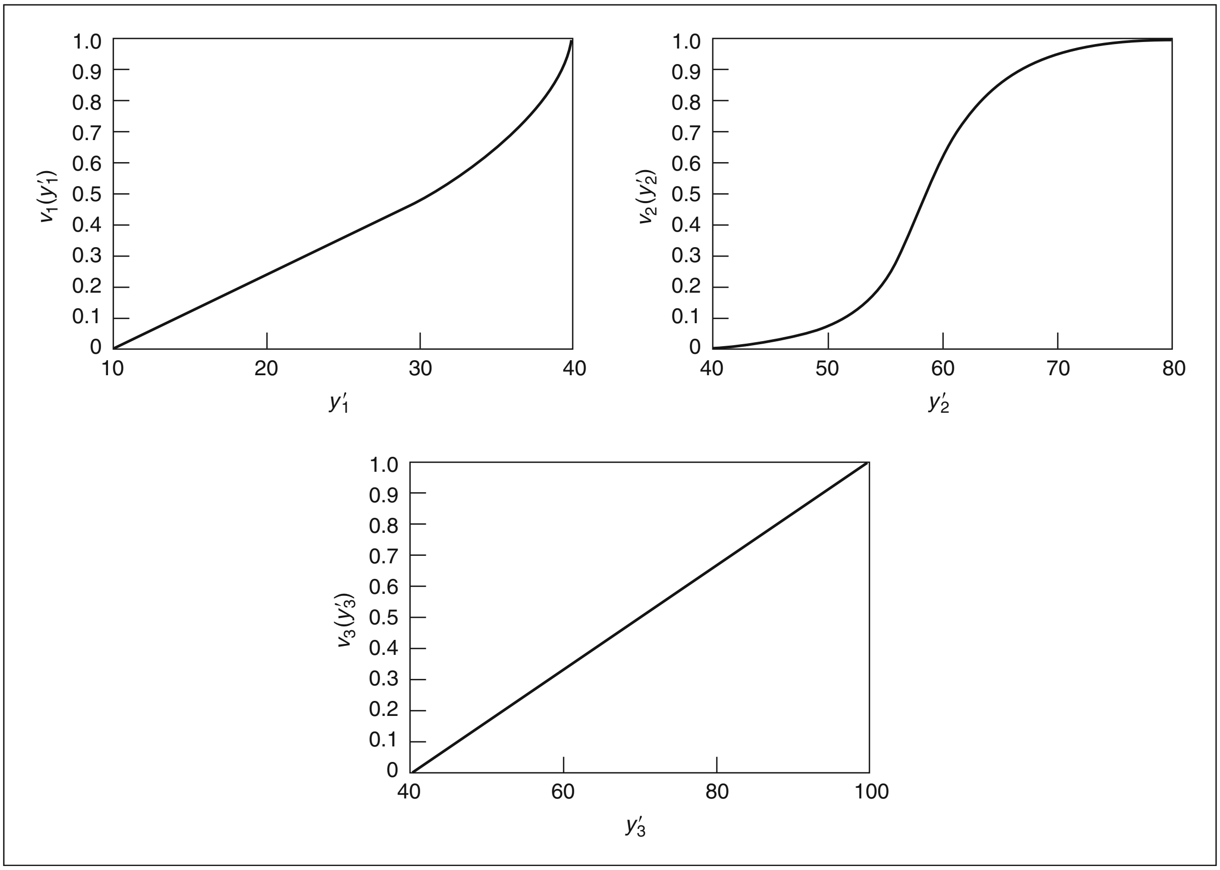
D. Summary

To put all the concepts together regarding value functions, we would like to conclude this section with a numerical example. You are using multi-attribute utility theory to analyze a two-alternative, three-attribute decision-making problem involving uncertainty. The alternatives (possible outcomes and probability distributions) are described below:

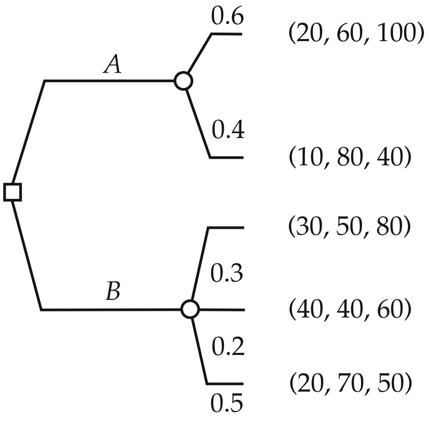
Alternative	Possible outcomes $(\mathbf{y}_1', \mathbf{y}_2', \mathbf{y}_3')$/probabilities p		
Site *A*	(20, 60, 100)/0.6	(10, 80, 40)/0.4	
Site *B*	(30, 50, 80)/0.3	(40, 40, 60)/0.2	(20, 70, 50)/0.5

Individual single-attribute utility functions, and the overall multiple-attribute utility function are represented by: $v(y_1', y_2', y_3') = 0.3v_1(y_1') + 0.5v_2(y_2') + 0.2v_3(y_3')$ and the graphical sketches as shown in Figure 5.17.

Figure 5.17 EXAMPLE UNIVARIATE UTILITY FUNCTIONS



(a) What site should the DM select and why?



$$
\begin{aligned}
v(A) &= 0.3[0.6v_1(20) + 0.4v_1(10)] + 0.5[0.6v_2(60) + 0.4v_2(80)] + 0.2[0.6v_3(100) + 0.4v_4(40)] \\
&= 0.3[0.6(0.23) + 0.4(0)] + 0.5[0.6(0.6) + 0.4(1)] + 0.2[0.6(1) + 0.4(0)] \\
&= 0.54
\end{aligned}
$$

$$\begin{aligned} v(B) &= 0.3[0.3v_1(30) + 0.2v_1(40) + 0.5v_1(20)] + 0.5[0.3v_2(50) + 0.2v_2(40) + 0.5v_2(70)] \\ &\qquad + 0.2[0.3v_3(80) + 0.2v_3(60) + 0.5v_3(50)] \\ &= 0.3[0.3(0.5) + 0.2(1) + 0.5(0.25)] + 0.5[0.3(0.1) + 0.2(0) + 0.5(0.95)] \\ &\qquad + 0.2[3(0.7) + 0.2(0.35) + 0.5(0.2)] \\ &= 0.48 \end{aligned}$$

Hence $A > B$, or Site A is preferred to Site B.

(b) What properties about the preferences of the DM did the decision analyst need to demonstrate in order to show that the form of the total utility-function is theoretically correct?

Following the discussion in Section VI-C directly above, one needs to show independence among all combinations of y_i's.

It can be seen that value functions can be used to rank order alternatives according to Equation 5.7. In the case where univariate utility functions are available and the exact form of the value function is known, as shown in the numerical example directly above, the ranking is cardinal. Thus alternative A is exactly 0.54/0.48 = 1.13 times more preferable to alternative B. A premise of such a statement is that the constituent attributes are independent. In the section immediately below, we will further discuss independence among y_i's in some detail.

VII. VALUE-FUNCTION MEASUREMENT STEPS

Should preference intensity be required, the precise form of the univariate utility functions and their aggregation into a value function are required, as already alluded to above. This involves the calibration coefficients, *k*s and *w*s. It also brings us face to face with the core of MCDM. A five-step process is prescribed to carry out this task (Zelany 1982):

1. Familiarize DM with the concepts and techniques of value function measurement.
2. Identify the appropriate value decomposition form, $v(\mathbf{y}')$.
3. Measure component value functions $v_i(y_i')$.
4. Determine the *k*s and *w*s.
5. Validate the consistency of $v(\mathbf{y}')$ against DM's observed rankings.

First and foremost, the DM and the stakeholders must be involved with the definition of value function, or what we have been referred to as the Z'-space. After all, the value function is supposed to reflect the DM's way of looking at the world. The simplest of all value functions has only linear terms, $v = w_1y_1' + w_2y_2'$, where the $v_i(y_i') = y_i'$. However, this simple form is the exception rather than the rule in real world applications. Should we be forced to decide between an additive versus multiplicative function, the discriminant is the type of independence between the attributes y_i's. An additive value function requires that the attributes be **preferentially independent,** while mutual utility independence is necessary in addition for **multiplicative utility** models. Limiting the value function to these two decomposition forms has the computational advantage of calibrating only in the order of q coefficients as mentioned.

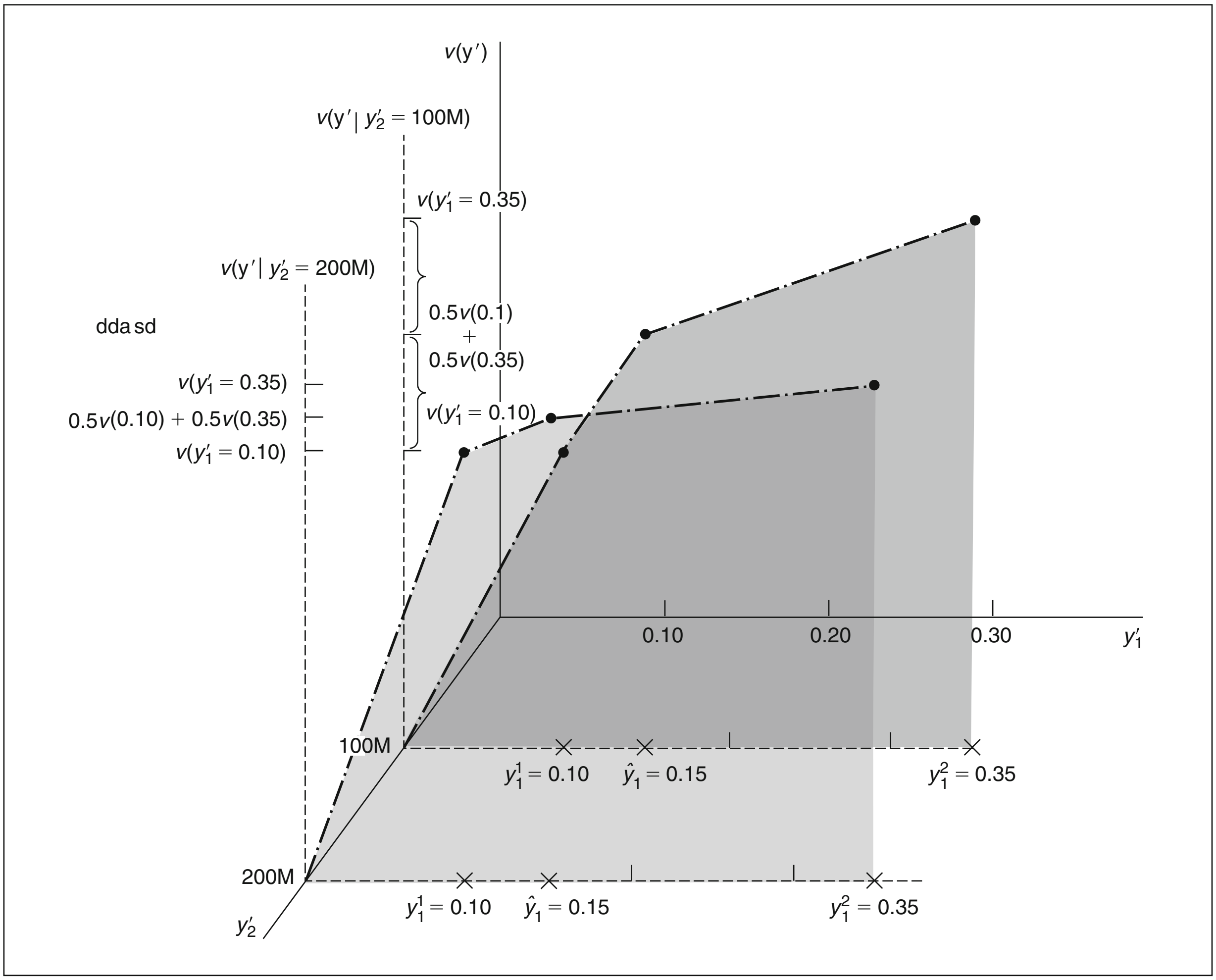
A. Preferential, Utility and Additive Independence

The pair of attributes y'_1 and y'_2 is preferentially independent of attribute y'_3 if the value tradeoff between y'_1 and y'_2 is not affected by a given level of y'_3. Formally stated, if $[(y_1^{\mathrm{I}}, y_2^{\mathrm{I}})\,|\,y_3'] > [(y_1^{\mathrm{II}}, y_2^{\mathrm{II}})\,|\,y_3']$, then $[(y_1^{\mathrm{I}}, y_2^{\mathrm{I}})\,|\,y_3''] > [(y_1^{\mathrm{II}}, y_2^{\mathrm{II}})\,|\,y_3'']$, where $y''_3 \mp y'_3$. Following the example on locating a plant among candidate states, labor availability, transportation, and tax advantages are considered. The value tradeoff between labor availability and transportation at two locations I and II may not depend on the tax advantage. In this case, we say that labor availability and transportation is preferentially independent of tax advantage, and location I is preferred to II in terms of labor-transportation considerations irrespective of tax differences. Preferential independence concerns ordinal preferences among attributes. Should all pairwise attributes pass this preferential independence test, it is conceivable then that the plant location problem could be modeled using a linear, additive value function. Simply stated, preferential independence means the tradeoffs between any two attributes are governed by the unique indifference curve between these two attributes regardless of the values of other attributes (Ang and Tang 1984). If attribute 1 is preferentially independent of attribute 2, and attribute 2 is preferentially independent of attribute 1, then attribute 1 is mutually preferentially independent of attribute 2. If a set of attributes $y'_1, \ldots, y'_n$ is mutually preferential independent, the decision maker's preferences can be represented by an additive value function. However, preferential independence is a necessary but not sufficient condition for an additive value function.

Utility independence, on the other hand, says the relative utility of y'_i remains the same regardless of other y'_js. In other words, the utility of each of the y'_is can be separately determined. Attribute y'_1 is utility independent of attribute y'_{i} when conditional preferences for lotteries on y'_1, given y'_2, do not depend on the particular level of y'_2. As an example let y'_1 be the anticipated percentage improvement owing to investments on sites 1 and 2 (with $y_1^1 = 35\%$ and $y_1^2 = 10\%$), also the probabilities of success are $P^1 = P^2 = 0.5$. Let y'_2 be the initial capital needed, with $y_2^1 = y_2^2 = \$100$ million; the certainty equivalent in this case is 15%. Now let the initial investment be $y_2^{1\prime} = y_2^{2\prime} = \$200$ million. If the certainty equivalent remains the same (in other words, it depends solely on the percentage improvements y_1^1 and y_1^2 and not on any fixed investment value y'_2) then attribute y'_1 would be utility independent of y'_2. This example is illustrated in Figure 5.18.

Utility independence is directional: y'_1 is utility-independent of y'_2 does not mean that y'_2 is utility independent of y'_1. Attributes $y'_1, y'_2, \ldots, y'_q$ are said to be mutually utility independent, if every subset of the attribute set is utility independent of its complement. As an example, let us consider this decomposable value function $v(y'_1, y'_2) = g[v_1(y'_1), v_2(y'_2)]$. Mutual utility independence is established among the attribute set $\{y'_1, y'_2\}$ *if and only if* y'_1 is utility independent of y'_2 and y'_2 utility-independent-of y'_1. To establish mutual utility independence is therefore complex, it requires formal lottery surveys to be conducted on uncertain outcomes in order to address the cardinal scale required of utility functions. Through these surveys, it is possible to measure the way utility changes over one

Figure 5.18 ILLUSTRATING UTILITY INDEPENDENCE



dimension, independent of all other attributes. These independent measurements can then be combined to give the multi-attribute utility function. An informal way of explaining mutual utility independence is to say that the shape of the utility function $v_i(y'_i)$ over $\mathbf{y}'$—whether risk-averse, risk-prone or risk-neutral—is the same irrespective of the level of all other attributes. An example of univariate functions of the same shape is shown in Figure 5.18. Notice here the two utility functions at y'_2 = 100 million dollars and y'_2 = 200 million look dissimilar, but they have the same indifference statement: 0.5(0.10) ⊗ 0.5(0.35) ~ 0.15. In other words, both curves indicate that the decision maker is indifferent between the certainty equivalent of 15 percent improvement and "achieving 35-percent-improvement with 50 percent chance" and getting only "10 percent improvement 50 percent of the time." (Notice that stating that two functions have the same shape is different from saying that two functions are strategically equivalent.) If we reverse y'_1 and y'_2 and are able to show utility independence between y'_2 and y'_1, then, y'_1 and y'_2 are mutually utility independent.

The extra amount of work associated with establishing mutual utility independence yields a much more useful metric. Instead of the mere rank order guarantee obtained from preferential independence, mutual utility independence quantifies the intensity of preference. To sum up the discussion on preferential and utility independence, let us review the basic concepts once again (Ang and Tang 1984). Effectiveness of an alternative is measured by several attributes via multi-attribute value functions. It is obvious that each of these attributes will require its respective unit of measurement, such as dollar costs, minutes of time, and parts-per-million of pollutants. Similar to Equation 5.7 the expected utility of an alternative $E(v(\mathbf{y}'))$ and the associated probability density function $P(\mathbf{y}')$ will, therefore, be multidimensional:

$$E(v(\mathbf{y}')) = \int_{y_1'} \ldots \int_{y_q'} v(y_1', \ldots, y_q')\, P(y_1', \ldots, y_q')\, dy_1' \ldots dy_q' \tag{5.10}$$

where y_1' to y_q' are random variables of the q respective attributes associated with each alternative. Determining these *joint* utility and density functions requires the evaluation of the conditional utility and probability functions. Moreover, these functions may have to be developed entirely or largely on the basis of subjective judgments and interviews, and they have to be performed for all the alternatives. This would generally be impractical if not impossible. Appropriate assumptions have been proposed in the above sections to exploit statistical independence among attributes, particularly preferential and utility independence. The assumptions of mutual preferential independence and mutual utility independence together imply that the joint utility function may be expressed as a function of the marginal (univariate) utility functions, namely the form $v(y_1', \ldots, y_q') = g[v_1(y_1'), \ldots, v_q(y_q')]$ where the $v(\cdot)$ is decomposable into function $g(\cdot)$ as defined by the following multiplicative expression (Keeney and Raiffa 1976)

$$kv(\mathbf{y}') + 1 = \prod_{i=1}^{q} [1 + kw_i v_i(y_i')] \tag{5.11}$$

Notice this follows Equation 5.6 in which a joint probability density function (PDF) is broken into univariate PDFs. Here $v(\mathbf{y}')$ and $v_i(y_i')$ are 0–1 ranged univariate utility functions (of the exponential form $v(y) = a + be^{-cy}$ for instance) as defined previously. The reader can check that $v(y')$ boils down to the familiar two- and three-dimensional multiplicative, decomposed form by setting $\mathbf{y}' = (y_1', y_2')$ and $\mathbf{y} = (y_1', y_2', y_3')$:

$$\begin{aligned}
&kv(\mathbf{y}') + 1 = [1 + kw_1v_1(y_1')][1 + kw_2v_2(y_2')][1 + kw_3v_3(y_3')]\\
&kv(\mathbf{y}') + 1 = 1 + kw_1v_1(y_1') + kw_2v_2(y_2') + kw_3v_3(y_3') + k^2w_1w_2v_1(y_1')v_2(y_2') + k^2w_2w_3v_2(y_2')v_3(y_3')\\
&\quad + k_2w_1w_3v_1(y_1')v_3(y_3') + k^3w_1w_2w_3v_1(y_1')v_2(y_2')v_3(y_3')v(\mathbf{y}') = w_1v_1(y_1') + w_2v_2(y_2') + w_3v_3(y_3')\\
&\quad + kw_1w_2v_1(y_1')v_2(y_2') + kw_2w_3v_2(y_2')v_3(y_3') + kw_1w_3v_1(y_1')v_3(y_3') + k^2w_1w_2w_3v_1(y_1')v_2(y_2')v_3(y_3')
\end{aligned}$$

After the univariate utility functions have been obtained, the function g may be determined by scaling $v_i(y_i')$ with respect to other utility functions such that they are consistent with one another and that $0 \le v(y) \le 1$. Consider the two-dimensional case (Ang and Tang 1984). It is obvious the outcomes $(y_1^{\text{Min}}, y_2^{\text{Min}})$ and $(y_1^{\text{Max}}, y_2^{\text{Max}})$ are the least and most desirable ones for the two-attribute utility function. In accordance with normal practice, we set $v(y_1^{\text{Min}}, y_2^{\text{Min}}) = 0$ and $v(y_1^{\text{Max}}, y_2^{\text{Max}}) = 1$. Then from Equation 5.11, the utility function with y_1' set at the least desirable state and y_2' at the most desirable state is given by

$$1 + kv(y_1^{\mathrm{Min}}, y_2^{\mathrm{Max}}) = [1 + kw_1\mathrm{v}_1(y_1^{\mathrm{Min}})][1 + kw_2v_2(y_2^{\mathrm{Max}})] = 1 + kw_2$$

or

$$w_2 = v(y_1^{\mathrm{Min}}, y_2^{\mathrm{Max}}) \tag{5.12}$$

and by symmetry, it can be shown that $w_1 = v(y_1^{\mathrm{Max}}, y_2^{\mathrm{Min}})$. Moreover, by substituting $v(y_1^{\mathrm{Max}}, y_2^{\mathrm{Max}})$ into Equation 5.11, we obtain

$$1 + kv(y_1^{\mathrm{Max}}, y_2^{\mathrm{Max}}) = [1 + kw_1\mathrm{v}_1(y_1^{\mathrm{Max}})]\ [1 + kw_2v_2(y_2^{\mathrm{Max}})]$$

or

$$1 + k = (1 + kw_1)(1 + kw_2) \tag{5.13}$$

Now the value of $v(y_1^{\mathrm{Max}}, y_2^{\mathrm{Min}})$ can be determined from a pair of indifferent lotteries as shown in Figure 5.12, where the payoff $\mathbf{y}^1$ is now $(y_1^{\mathrm{Max}}, y_2^{\mathrm{Max}})$, $\mathbf{y}^2$ is $(y_1^{\mathrm{Min}}, y_2^{\mathrm{Min}})$, and $Pv(\mathbf{y}^1) + (1 - P)v(\mathbf{y}^2)$ is set at $(y_1^{\mathrm{Max}}, y_2^{\mathrm{Min}})$. Suppose the decision maker is indifferent between alternatives I and II at probability $P_1 = v(\hat{\mathbf{y}}_1) = v(y_1^{\mathrm{Max}}, y_2^{\mathrm{Min}})$; then from Equation 5.12 $w_1 = P_1$, and $w_2 = v(\hat{\mathbf{y}}_2) = v(y_1^{\mathrm{Min}}, y_2^{\mathrm{Max}}) = P_2$ (by symmetry). From Equation 5.13

$$k = (1 - w_1 - w_2)/w_1w_2 = (1 - P_1 - P_2)/P_1P_2$$

Hence the two-attribute utility function is calibrated to be

$$\begin{aligned} v(y'_1, y'_2) &= w_1v_1(y'_1) + w_2v_2(y'_2) + kw_1w_2v_1(y'_1)v_2(y'_2) \\ &= P_1v_1(y'_2) + P_2v_2(y'_2) + (1 - P_1 - P_2)v_1(y'_1)v_2(y'_2) \end{aligned}$$

The multiplicative utility function above is the most general representation of a multi-attribute utility function in consideration for the efficiency with which such functions can be calibrated. We recall that when $k = 0$, the multiplicative function reduces to the simple additive form, with $w_1 + w_2 \cdots + w_q = 1$. We say that the y_is exhibit **additive independence** in this case.

Suppose attributes 1 and 2 are mutually utility independent, a DM's utility function exhibits additive independence if the DM is indifferent between the following alternatives I and II.

Alternative I: $(y_1^{\mathrm{Max}}, \mathrm{y}_2^{\mathrm{Max}})$ with probability 0.5 vs.
$(y_1^{\mathrm{Min}}, y_2^{\mathrm{Min}})$ with probability 0.5

Alternative II: $(y_1^{\mathrm{Max}}, \mathrm{y}_2^{\mathrm{Min}})$ with probability 0.5 vs.
$(y_1^{\mathrm{Min}}, y_2^{\mathrm{Max}})$ with probability 0.5

Consistent with our notation, y_1^{Max} means the best of attribute 1, and y_1^{Min} means the worst of attribute 1. This is similar for attribute 2.

Let us illustrate this property with our familiar example of siting a manufacturing plant. Suppose attribute 1 is labor availability and attribute 2 is accessibility. One does not know precisely about the degree of labor availability and accessibility at sites I vs. II, inasmuch as we are projecting 10 years into the future when the plant is actually built and ready for operation. To some extent, the two uncertain attributes are substitutes of one another as illustrated below.

Site I: (Excellent labor availability, excellent accessability) with probability 0.5 vs. (Poor labor availability, poor accessability) with probability 0.5

Site II: (Excellent labor availability, poor accessability) with probability 0.5 vs. (Poor labor availability, excellent accessability) with probability 0.5

When the DM is equally inclined toward locations I vs. II, each with uncertain levels of attributes 1 and 2, then one can say that labor availability is additively independent of accessibility. On the other hand, if the DM has a clear preference for one site or the other, then additive independence cannot hold. In the former case, preferences over lotteries depend only on the marginal distribution of labor availability (or accessibility), and do not depend on the joint distribution of the possible values of the two attributes. The intuition behind additive independence is that, in assessing uncertain outcomes over both attributes, we only have to look at one attribute at a time. It does not matter what the other attribute values are in the uncertain outcomes.

If attributes 1 and 2 are mutually utility independent and the DM's utility function exhibits additive independence, only the additive terms of the univariate utility functions apply, with the cross term v_1v_2 dropped. Mathematically, it is easy to show that $k = 0$ in a two-attribute utility function, eliminating the cross/multiplicative term. Suppose the component univariate utility function have been properly scaled between 0 and 1, or $v(y_1^{\text{Min}}) = v(y_2^{\text{Min}}) = 0$ and $v(y_1^{\text{Max}}) = v(y_2^{\text{Max}}) = 1$. Then

$$v(y_1^{\text{Max}}, y_2^{\text{Max}}) = w_1v_1(y_1^{\text{Max}}) + w_2\, v_2(y_2^{\text{Max}}) + k\, w_1w_2\, v_1(y_1^{\text{Max}})\, v_2(y_2^{\text{Max}}) = w_1 + w_2 + k\, w_1w_2$$
$$v(y_1^{\text{Min}}, y_2^{\text{Min}}) = w_1v_1(y_1^{\text{Min}}) + w_2\, v_2(y_2^{\text{Min}}) + k\, w_1w_2\, v_1(y_1^{\text{Min}})\, v_2(y_2^{\text{Min}}) = 0$$
$$v(y_1^{\text{Max}}, y_2^{\text{Min}}) = w_1v_1(y_1^{\text{Max}}) + w_2\, v_2(y_2^{\text{Min}}) + k\, w_1w_2\, v_1\, (y_1^{\text{Max}})\, v_2(y_2^{\text{Min}}) = w_1$$
$$v(y_1^{\text{Min}}, y_2^{\text{Max}}) = w_1v_1(y_1^{\text{Min}}) + w_2\, v_2(y_2^{\text{Max}}) + \ k\, w_1w_2\, v_1(y_1^{\text{Min}})\, v_2(y_2^{\text{Max}}) = w_2$$

Additive independence implies that

$$0.5(w_1 + w_2 + k\, w_1w_2) + 0.5\ (0) = 0.5\ (w_1) + 0.5\ (w_2),\ \text{or}\ k\, w_1w_2 = 0$$

To the extent that $w_1w_2 \neq 0$, k must then be zero valued.

We will now illustrate the procedure of calibration in the following numerical examples, namely steps 2 through 4 of the five-step value function measurement process.

B. Examples of Utility Function Calibration

We will illustrate the calibration of a multi-attribute utility function via two examples. The first will show an additive function where the weights w_i are to be determined. The second will show a multiplicative function where both the weights w_i and the scaling constant k are to be determined.

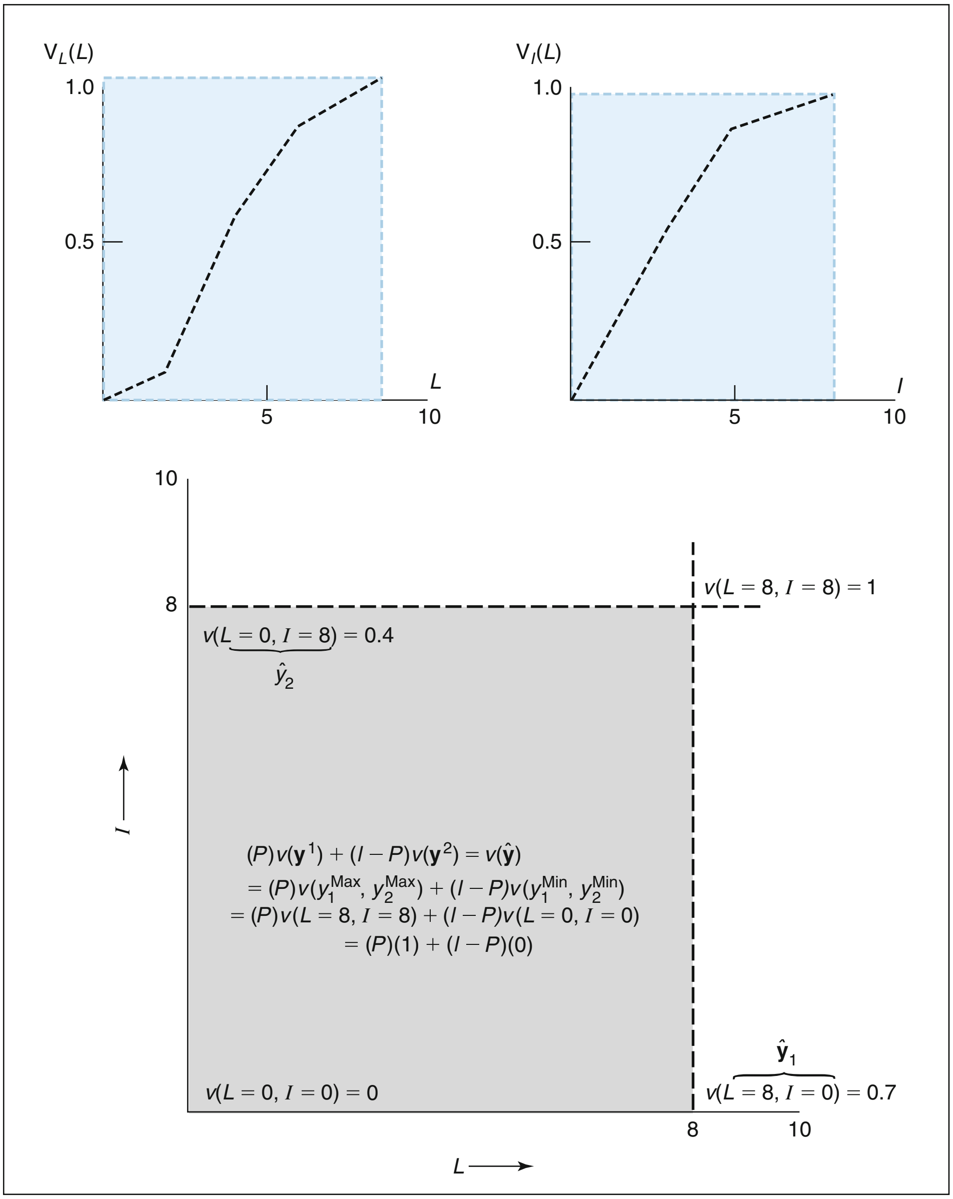
Example 1: Determination of Weights

Suppose for the time being, the component univariate utility functions have been determined as $v_1(y'_1) = 3/2\ y'_1 - 1/2\ y'^2_1$, $v_2(y'_2) = 3/4\ y'_2 - 1/8\ y'^2_2$, and $v_3(y'_3) = y'_3 - 1/4\ y'^2_3$. Also the value function is determined to be additive, meaning that $k = 0$, and $\Sigma_i w_i = 1$. An interview with the decision maker yields the indifference relationships when the multivariate value function assumes different attribute levels for y'_1, y'_2 and y'_3: (0, 1, 1) ~ (1, 0, 1) and (1, 1, 1) ~ (0, 2, 2). Now substitute the indifference results from the interview into the composite value function, or $v[(0, 1, 1)] = v[(1, 0, 1)]$ We solve for the w_is: $5/8\ w_2 + 3/4\ w_3 = w_1 + 3/4\ w_3$ or $w_1 = 5/8\ w_2$. Similarly, setting $v[(1, 1, 1)] = v[(0, 2, 2)]$ yields $w_1 + 5/8\ w_2 + 3/4\ w_3 = w_2 + w_3$. Solving these two equations together with $w_1 + w_2 + w_3 = 1$, we yield $(w_1, w_2, w_3) = (5/21, 8/21, 8/21)$ (Yu (1985)]. ■

Example 2: Two-Attribute Utility Function Calibration

For a large urban area, landfill (*L*) and incinerators (*I*) are alternatives for solid-waste disposal. The univariate utility functions for each option are shown in Figure 5.19,

Figure 5.19 TWO-ATTRIBUTE UTILITY-FUNCTION CALIBRATION EXAMPLE



where up to eight landfills or incinerators are considered (de Neufville 1990). Lotteries unveil these indifference relationships:

$$0.7(L = 8, I = 8) \otimes 0.3(L = 0, I = 0) \sim (L = 8, I = 0)$$
$$0.4(L = 8, I = 8) \otimes 0.6(L = 0, I = 0) \sim (L = 0, I = 8).$$

The first line says the following: The population is indifferent between a lottery at "70-percent probability of having all the 8 landfills and 8 incinerators built, and 30-percent none at all," vis-a-vis "building all 8 landfills and zero incinerator." The second line shows the indifference between "40-percent having all facilities built" vis-a-vis 60-percent "all incinerators only." From these lotteries, one can conclude

$$v(L = 8, I = 0) = 0.7\, v(L = 8, I = 8) + 0.3\, v(L = 0, I = 0) = (0.7)(1) + (0.3)(0) = 0.7$$
$$v(L = 0, I = 8) = 0.4\, v(L = 8, I = 8) + 0.6\, v(L = 0, I = 0) = (0.4)(1) + (0.6)(0) = 0.4.$$

Thus $w_1 = 0.7$ and $w_2 = 0.4$. $k = (1 - 0.7 - 0.4)/(0.7)(0.4) = -0.1/(0.7)(0.4) = 0.357$ according to Equations 5.12 and 5.13 respectively. The two-attribute utility function now looks like

$$v(L, I) = 0.7\, v_L(L) + 0.4\, v_I(I) - 0.1\, v_L(L)\, v_I(I). \blacksquare$$

Example 3: Determination of 3-Attribute Utility Function

A client has asked you to help him compare three commercial land development projects and choose the best one. Each project is to develop a shopping center. The client can fund only one of these efforts and must begin development as soon as possible. Each project has been evaluated by the client in terms of cost, time to completion, and effectiveness for each of two possible growth scenarios. However, the type of growth profile is not known. The growth profile will be one of two types—high or low—and each is equally likely.

The performance of the three projects for a high and low growth profile is determined in Table 5.1. You decide to construct a multi-attribute utility function v to help evaluate the three projects after obtaining the following information from the client. The three attributes—cost, time to completion, and effectiveness—have the following ranges:

Table 5.1 PERFORMANCE OF PROJECTS FOR HIGH- AND LOW-GROWTH PROFILES[4]

Project	High			Low		
	Cost y_1	Time y_2	Effect y_3	Cost y_1	Time y_2	Effect y_3
A	20	30	40	25	20	40
B	30	10	50	30	15	45
C	25	20	60	30	20	50

cost	20–40 (less is preferred)
time to completion	10–30 (less is preferred)
effectiveness	40–60 (more is preferred)

Armed with this information, you solicit the following indifference and lottery data from the client, where the three entries in parenthesis refer to cost (C), time to completion (T), and effectiveness (E) respectively.

Indifference set 1:	$(30, 20, 60) \sim (35, 17, 60) \sim (20, 25, 60)$
	$(30, 20, 40) \sim (35, 17, 40) \sim (20, 25, 40)$
Indifference set 2:	$(40, 20, 50) \sim (40, 25, 60) \sim (40, 17, 45)$
	$(20, 20, 50) \sim (20, 25, 60) \sim (20, 17, 45)$
Indifference set 3:	$(30, 30, 50) \sim (25, 30, 45) \sim (35, 30, 55)$
	$(30, 10, 50) \sim (25, 10, 45) \sim (35, 10, 55)$
Indifference set 4:	$(40, 30, 60) \sim (20, 30, 40)$
Indifference set 5:	$(40, 20, 40) \sim (20, 30, 40)$
	$(40, 20, 40) \sim (40, 30, 60)$
Lottery set 6:	$0.8(40, 30, 40) \otimes 0.2(20, 10, 60) \sim (20, 30, 40)$

Here, $\mathbf{y}^i \sim \mathbf{y}^j$ means project i is indifferent from project j, and $P \otimes Q$ stands for a lottery between the outcomes P and Q. We have also assumed that the utility function of the DM is multiplicative with constant parameters w_1, w_2, w_3, and k. Obviously, such an assumption needs to be justified as will be shown later. It is prudent to start with a multiplicative form since the additive form can be thought of as a special case when $k = 0$.

Then the following information is used to specify single-attribute utility functions using the quartile method. For example, $0.5(20) \otimes 0.5(40)$ means a lottery in which there is a 50–50 chance of obtaining a score of 20 or 40.

Set 7: Lotteries over C, given $T = 10$, $E = 60$ and $T = 30$, $E = 40$,

$$0.5(20) \otimes 0.5(40) \sim 30$$
$$0.5(30) \otimes 0.5(40) \sim 35$$
$$0.5(30) \otimes 0.5(20) \sim 25.$$

Set 8: Lotteries over T, given $C = 40$, $E = 40$ and $C = 20$, $E = 60$,

$$0.5(10) \otimes 0.5(30) \sim 20$$
$$0.5(20) \otimes 0.5(30) \sim 25$$
$$0.5(20) \otimes 0.5(10) \sim 15$$

Set 9: Lotteries over E, given $C = 40$, $T = 30$ and $C = 20$, $T = 10$,

$$0.5(40) \otimes 0.5(60) \sim 50$$
$$0.5(50) \otimes 0.5(40) \sim 45$$
$$0.5(50) \otimes 0.5(60) \sim 55.$$

Notice the above lotteries show utility independence. In other words, preferences for lotteries involving different levels of costs do not depend on the levels of time and effectiveness. Preferences for lotteries involving different levels of time do not depend on the levels of cost and effectiveness. Finally, preferences for lotteries involving different levels of effectiveness do not depend on the levels of cost and time.

Now what should the client do regarding project selection? Specifically,

(a) Which data set establishes that preferences in *T*-*C* space are preferentially independent of *E*?

Set 1 establishes that {*C*, *T*} is preferentially independent of {*E*}, inasmuch as effectiveness level is held constant in this set.

(b) Which set establishes that preference in *T*-*E* space are preferentially independent of *C*?

Set 2 establishes that {*T*, *E*} is preferentially independent of {*C*}.

(c) Which set establishes that *E* is utility independent of *C* and *T*?

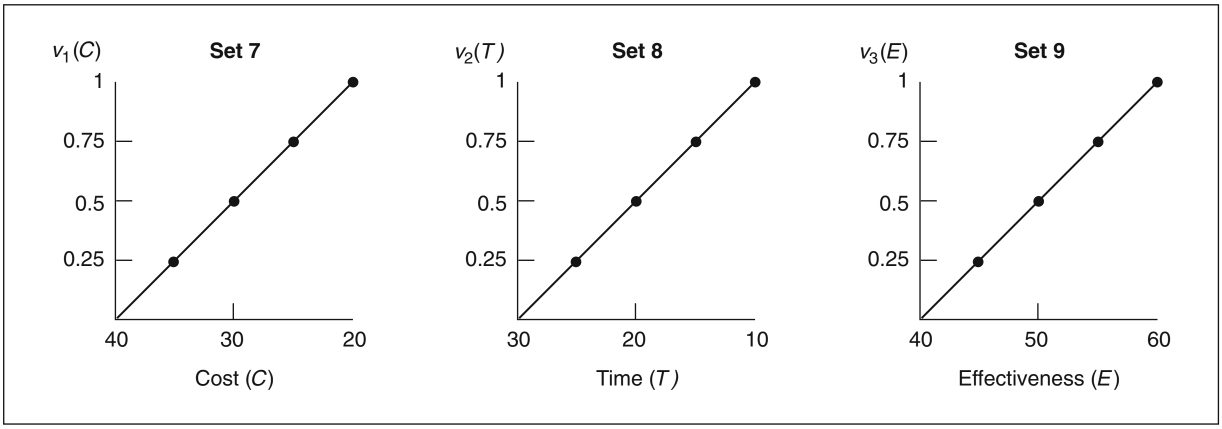
Set 9 establishes that {*E*} is utility independent of {*C*, *T*}.

(d) Draw and label the single-attribute utility functions for *C*, *T* and *E*. Specify the set that was used in constructing each of the three utility functions.

The single-attribute utility functions are shown to be all risk-neutral in Figure 5.20. To summarize, mutual preferential and mutual utility independence are established for these attributes:

Set 1 establishes that {*C*, *T*} is preferentially independent of {*E*} and {*T*, *C*} preferentially independent of {*E*};
Set 2 establishes that {*T*, *E*} is preferentially independent of {*C*} and {*E*, *T*} preferentially independent of {C};
Set 3 establishes that {*C*, *E*} is preferentially independent of {*T*} and {*E*, *C*} preferentially independent of {*T*};
Set 7 establishes that {*C*} is utility independent of {*E*, *T*};

Figure 5.20 UNIVARIATE UTILITY FUNCTIONS FOR COST, TIME, AND EFFECTIVENESS



Set 8 establishes that {T} is utility independent of {C, E};
Set 9 establishes that {E} is utility independent of {C, T}.

Since cost, time to completion, and effectiveness are mutually utility independent, this means that the multi-attribute utility function is in fact multiplicative. Remember that mutual utility independence is a necessary and sufficient condition for a multiplicative functional form.

(e) Calculate the weights and scaling constant for the multiplicative utility-function. Show and explain your work.

According to set 4, $v(40, 30, 60) = v(20, 30, 40)$ or $v(y_1^{\text{Min}}, y_2^{\text{Min}}, y_3^{\text{Max}}) = v(y_1^{\text{Max}}, y_2^{\text{Min}}, y_3^{\text{Min}})$, which means $w_3 = w_1$ according to the three-attribute expansion of Equation 5.11. From set 5, $v(40, 20, 40) = v(20, 30, 40)$ or $v(y_1^{\text{Min}}, y_2^{\text{between}}, y_3^{\text{Min}}) = v(y_1^{\text{Max}}, y_2^{\text{Min}}, \mathrm{y}_3^{\text{Min}})$. Hence $w_1 = w_2 v_2(20)$, or $w_1 = 0.5w_2$. From set 5 again, $v(40, 20, 40) = v(40, 30, 60)$ or $v(y_1^{\text{Min}}, y_2^{\text{between}}, y_3^{\text{Min}}) = v(y_1^{\text{Min}}, y_2^{\text{Min}}, y_3^{\text{Max}})$. Therefore $w_2 v_2(20) = w_3$, or $w_3 = 0.5w_2$. From set 6,

$$(20, 30, 40) \sim \bigcirc \begin{array}{l} \diagup\,\overline{0.8}\quad (40, 30, 40) \\ \diagdown\,\underline{0.2}\quad (20, 10, 60) \end{array} \tag{5.14}$$

or

$$(y_1^{\text{Max}}, y_2^{\text{Min}}, y_3^{\text{Min}}) \sim \bigcirc \begin{array}{l} \diagup\,\overline{0.8}\quad (y_1^{\text{Min}}, y_2^{\text{Min}}, y_3^{\text{Min}}) \\ \diagdown\,\underline{0.2}\quad (y_1^{\text{Max}}, y_2^{\text{Max}}, y_3^{\text{Max}}) \end{array} \tag{5.15}$$

Hence $v(y_1^{\text{Max}}, y_2^{\text{Min}}, y_3^{\text{Min}}) = w_1 = 0.8(0) + (0.2)(1) = 0.2$. This leads to $w_2 = 0.4$ and $w_3 = 0.2$. Substituting these values of w into the three-attribute expansion of Equation 5.11 when $v(\mathbf{y}^{\text{Max}}) = 1$ and $v_i(y_i^{\text{Max}}) = 1$, $1 = w_1 + w_2 + w_3 + k(w_1w_2 + w_1w_3 + w_2w_3) + k^2(w_1w_2w_3)$ yields $0.016k^2 + 0.20k - 0.2 = 0$ or $k = 0.9307$. The multivariate function now assumes the form $v(y'_1, y'_2, y'_3) = 0.2v_1(y'_1) + 0.4v_2(y'_2) + 0.2v3(y'_3) + 0.0745v_1(y'_1)v_2(y'_2) + 0.0372v_1(y'_1)v_3(y'_3) + 0.0745v_2(y'_2)v_3(y'_3) + 0.0139v_1(y'_1)v_2(y'_2)v_3(y'_3)$.

It should be noted that there is a numerical relationship between the scaling factor k and the weights w_is. $\Sigma_i w_i < 1$ implies $k > 0$, $\Sigma_i w_i > 1$ implies $-1 < k < 0$, and as already mentioned, when $\Sigma_i w_i = 1$, $k = 0$. The weights w_i represent the multi-attribute utility of y'_i when y'_i is at its best level and all other y_js ($j \neq i$) at their worst. In order to calibrate a multi-attribute utility function, a survey needs to include enough questions to assess the indifference relationships between utilities of special combinations of the criteria levels $(y'_1, \ldots, y'_q)$, whereby sufficient equations are obtained for the determination of w_i.

(f) Calculate the expected utility of each development project.

Given the high- and low-growth profiles are equally likely, the expected utilities of the three projects A, B, and C can be readily calculated:

$E(v(\mathbf{y}^A)) = 0.5v(20, 30, 40) + 0.5v(25, 20, 40) = (0.5)(0.2) + (0.5)(0.3779) = 0.2890.$
$E(v(\mathbf{y}^B)) = 0.5v(30, 10, 50) + 0.5v(30, 15, 45) = (0.5)(0.6873) + (0.5)(0.4979) = 0.5926.$
$E(v(\mathbf{y}^C)) = 0.5(v(25, 20, 60) + 0.5v(30, 20, 50) = (0.5)(0.6483) + (0.5)(0.4482) = 0.5482.$

(g) Which project should the client choose if your utility function does in fact represent his/her preferences? Explain.

To maximize expected utility, the client should choose project *B*. ■

Readers interested in details of multi-attribute utility theory can find further reading in Keeney and Raiffa (1976), Zelany (1982), and Seo and Sakawa (1988). Empirical calibration procedures are outlined in de Neufville (1990) and Bana e Costá (1990).

C. Validation

To illustrate the rest of the five-step value function measurement process, suppose the value function has a simple additive form $v(y) = w_1v_1(y_1) + \cdots + w_qv_q(y_q)$ and a DM has determined a weight ratio to show the importance for every possible pair of criteria:

criteria	1	2	3
1	w_1/w_1	w_1/w_2	w_1/w_3
2	w_2/w_1	w_2/w_2	w_2/w_3
3	w_3/w_1	w_3/w_2	w_3/w_3

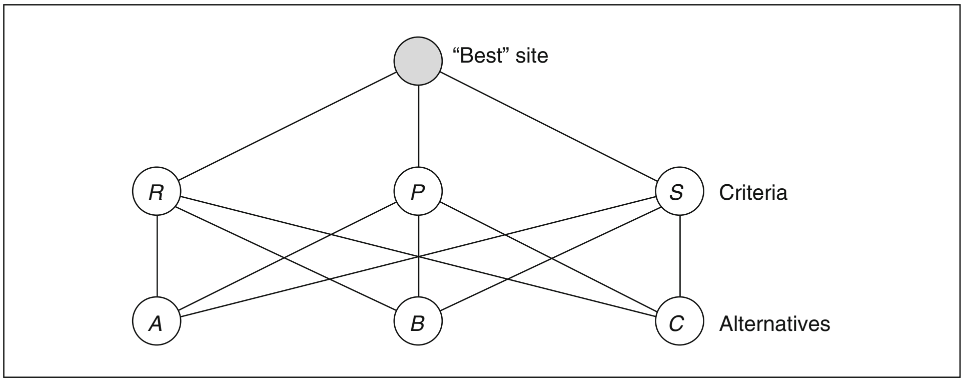
Thus $w_{12} = w_1/w_2$ shows the relative importance of criterion 1 against criterion 2. In general $w_{ij} = w_i/w_j$ in the matrix W = $[w_{ij}]_{qxq}$. An interview matrix **W** is *consistent* if $w_{ij} = w_{ji}^{-1}$ and $w_{ij} = w_{ik}w_{kj}$, or criterion *i* is preferred to *j* the same way as *j* is preferred to *i*. This consistency in part validates a value function as specified in step 5 of the five-step process. Now given the pair-wise-comparison ratios w_{ij}, the weights w_i should satisfy the following set of equations:

$$
\begin{aligned}
w_{11}w_1 + w_{12}w_2 + w_{13}w_3 &= q'w_1 \\
w_{21}w_1 + w_{22}w_2 + w_{23}w_3 &= q'w_2 \\
w_{31}w_1 + w_{32}w_2 + w_{33}w_3 &= q'w_3
\end{aligned}
$$

q' is the eigenvalue and **w** is the eigenvector in the above equation set $\mathbf{Ww} = q'\mathbf{w}$ or $(\mathbf{W} - q'\mathbf{I})\mathbf{w} = \mathbf{0}$.

Notice q' can be uniquely determined, considering we have a fourth equation $w_1 + w_2 + w_3 = 1$. In the above example, for instance, $q' = 3$ if everything is consistent. If an interview with the DM yields a matrix **W**′ (instead of **W**), and the eigenvalue is 3.5, the weights by the DM are inconsistent. The bigger the eigenvalue is, the larger the inconsistency. The same set of simultaneous equations can be defined for analyzing the univariate value functions $v_i(\mathbf{y}'_i)$, where the weight eigenvector $\mathbf{w} = (w_1, \ldots, w_q)^T$ is now replaced by the scores of alternatives *j* on criterion *i* $\mathbf{v}_i = (v_i^1, \ldots, v_i^{|J|})^T$, where $|J|$ is the number of alternatives (in lieu of *q*, the number of criteria), and $0 \leq v_{ij} \leq 1$ (just like $0 \leq w_i \leq 1$). An example is shown later in which a three alternatives, *A*, *B*, and *C*, are evaluated in terms of risk, performance, and schedule compliance. Figure 5.21 illustrates this point graphically, particularly where the univariate risk-criterion utility function v_R is expressed in terms of the utilities of the three alternatives *A*, *B*, and *C*: $v_R = w_R^Av_R^A + w_R^Bv_R^B + w_R^Cv_R^C$. Here $(v_R^A, v_R^B, v_R^C)^T$ is

Figure 5.21 ANALYTIC HIERARCHY PROCESS EXAMPLE



the eigenvector to be determined, and w_R^A, w_R^B and w_R^C have been obtained from the interview. We will step through these calculations subsequently. Suffice to say here that when contrasting this approach with the lottery method described previously, one can see that the current approach tends to assume additive decomposition all the way, not only in the multi-attribute value function between risk (R), performance (P), and schedule compliance (S), $v = w_R v_R + w_P v_P + w_S v_S$, but also in determining the univariate functions v_R, v_P and v_S of these criteria. Instead of estimating a full univariate utility function, a point estimate is made. Saaty's (1980) widely disseminated *The Analytic Hierarchy Process* is based on the above concepts.

In general, for a system of equations such as $(\mathbf{W} - q'\mathbf{I})\mathbf{w} = \mathbf{0}$, if $w_{ii} = 1$, then $\Sigma_{k=1}^{a} q'_k = q$ for all eigenvalues q'_k that satisfy the equations. The eigenvalues q'_k constitute a measure of consistency of the AHP. If the answers of DM are totally consistent, the principal eigenvalue $q'_{\text{Max}} = q$ and $q'_k = 0$ for all other ks. Should one perturb the perfect entries w_{ij}s by a small amount (which often occurs in actual interviews with decision makers), the eigenvalues $q'(-\infty \leq q \leq \infty)$ change by small amounts also. Small variations in w_{ij} keep the q'_{Max} close to q and the rest close to 0, some of which may be slightly less than 0. The result $q'_{\text{Max}} \geq q$ always holds. A consistency index (CI), $(q'_{\text{Max}} - q)/(q - 1)$, will measure the closeness to consistency. In general, a CI less than 0.1 is considered acceptable. Notice again that the process consists of normalizing w's, or setting the equation $\Sigma_i w_i = 1$, perturbations on the $\mathbf{W}$ matrix will yield $q' \geq q$ even for a perfectly symmetrical $\mathbf{W}$ matrix.

Example

The **analytic hierarchy process** (AHP) is used to assess hazardous facility siting. The best site is evaluated with respect to the risk (R), performance (P), and schedule-of-completion (S), resulting in the following tradeoff weights $[w_{ij}]$:

Best site:	risk	perf	sched
risk	1	1/3	2
perf	3	1	3
sched	1/2	1/3	1

Similarly, the three candidate sites *A*, *B* and *C* are compared among themselves with respect to the three criteria: risk, performance, and schedule, resulting in the weights $[w_{ij}]$:

risk	*A*	*B*	*C*
A	1	1	2
B	1	1	2
C	1/2	1/2	1

perf	*A*	*B*	*C*
A	1	3	9
B	1/3	1	1/7
C	1/9	7	1

sched	*A*	*B*	*C*
A	1	3	1/9
B	1/3	1	1/7
C	9	7	1

The graphical representation of this problem is shown in Figure 5.21, which has a two-level hierarchy, with $\mathbf{w}$ to be determined in the first level, and $\mathbf{v}_i$ the second level.

(a) Compute the weight eigenvector $\mathbf{w} = (w_R, w_P, w_S)^T$ and eigenvalue q'_{Max} for the best site.

$$\begin{aligned}
1\,w_R + 1/3\,w_P + 2\,w_S &= q'w_R\\
3\,w_R + 1\,w_P + 3\,w_S &= q'w_P\\
1/2w_R + 1/3\,w_P + 1\,w_S &= q'w_S\\
w_R + w_P + w_S &= 1
\end{aligned}$$

Here $\mathbf{w} = (0.249, 0.593, 0.158)$, $q'_{\text{Max}} = q' = 3.053$ and CI $= 0.026$.

(b) Now write a composite value-function of additive form to include all the component univariate value functions of risk (*R*), performance, (*P*) and schedule (*S*).

$$v = 0.249v_R + 0.593v_P + 0.158v_S$$

(c) Compute the eigenvector $\mathbf{v}_i = (\mathbf{v}_i^A, \mathbf{v}_i^B, \mathbf{v}_i^C)$ and eigenvalue for each of the criteria $i = R, P,$ and S.

$$\begin{aligned}
1\,v_R^A + 1\,v_R^B + 2\,v_R^C &= q'_R v_R^A\\
1\,v_R^A + 1\,v_R^A + 2\,v_R^C &= q'_R v_R^A\\
1/2v_R^A + 1/2\,v_R^A + 1\,v_R^C &= q'_R v_R^A\\
v_R^A + v_R^B + v_R^C &= 1
\end{aligned}$$

Hence $\mathbf{v}_R = (v_R^A, v_R^B, v_R^C)^T = (0.4, 0.4, 0.2)^T$, $q'_{\text{Rmax}} = 3$, and CI $= 0$.

$$\begin{aligned}
1\,v_P^A + 3\,v_P^B + 9\,v_P^C &= q'_P v_P^A\\
1/3\,v_P^A + 1\,v_P^B + 1/7v_P^C &= q'_P v_P^B\\
1/9\,v_P^A + 7\,v_P^B + 1\,v_P^C &= q'_P v_P^C\\
v_P^A + v_P^B + v_P^C &= 1
\end{aligned}$$

It follows that $\mathbf{v}_P = (v_P^A, v_P^B, v_P^C)^{\mathrm{T}} = (0.701, 0.084, 0.215)^T$, $q'_{\text{Pmax}} = 4.12$, and CI $= 0.56$.

$$\begin{aligned}
1v_S^A + 3v_S^B + 1/9\,v_S^C &= q'_S v_S^A\\
1/3v_S^A + 1v_S^B + 1/7\,v_S^C &= q'_S v_S^B\\
9v_S^A + 7v_S^B + 1\,v_S^C &= q'_S v_S^C\\
v_S^A + v_S^B + v_S^C &= 1
\end{aligned}$$

Therefore $\mathbf{v}_S = (v_S^A, v_S^B, v_S^C)^T = (0.138, 0.072, 0.79)^T$, $q'_{S\,\text{Max}} = 3.205$, and CI =0.103.

(d) Based on the composite value-function defined in (b) and the eigenvectors $\mathbf{v}_i = (v_i^A, v_i^B, v_i^C)^T$ computed in (c), rank order the preference among sites *A*, *B*, and *C*.

$$v^A = 0.249v_R^A + 0.593v_P^A + 0.158v_S^A = 0.537$$
$$v^B = 0.249v_R^B + 0.593v_P^B + 0.158v_S^B = 0.161$$
$$v^C = 0.249v_R^C + 0.593v_P^C + 0.158v_S^C = 0.302.$$

Therefore site *A* is preferred to *C*, which is in turn preferred to *B*. Notice here that we need not determine the precise form of the univariate utility functions v_R, v_P, and v_S. Only point estimate v_i's are necessary.

(e) Based on the eigenvalues in (c) above—q_R, q_P, and q_S—comment on the consistency of the DM and hence the validity of the rank order derived in (d).

With the exception of the risk criterion, the interviews yield rather inconsistent result, with the performance inconsistence and schedule inconsistency exceeding the CI maximum of 0.1 set by Saaty (1980). One can also argue that the schedule consistency is marginally acceptable. A poor consistency index reflects a number of problems. Here it may reflect the validity of the DM's responses during the interview. The consistency index also measures the independence among criteria, a concept somewhat parallel to that of preferential independence in multi-attribute utility theory. In this case, an overall CI of 0.026 suggests that the three criteria—risk, performance, and schedule—appear to be independent of each other. Too high a set of CIs can put into question the reliability of the ranking among alternatives. If the CIs are deemed unacceptable, the analyst needs redefine the criterion and to conduct the performance interview again to obtain a more reliable ranking.

The final step in the five-step process also calls for field validation of the rank order obtained above, which is a normal conclusion to value function modeling. Unless the ranking obtained from this set of value functions agrees with the observed ones, the process is not complete and more iterations through the five steps is required. Even though this is understood, one can be amazed at the number of applications where this last validation step is not carried out. ■

In general, multi-attribute utility analysis is a useful tool for decision making. However, the key lies in the conduct of the interviews with DM's. Pairwise comparisons and lottery questions typically are cumbersome to administer, which discounts to a large extent the usefulness of these techniques (Islam 1996). While there are constant debates over the correct theoretical underpinnings, it appears that the ultimate test is the ability of the procedure to reproduce and predict the DM's ranking simply and consistently. For example, the concept of strategic equivalence allows an additive value-function to replace a more complex one for ranking alternatives (see Section IV-A of the current chapter). If an ordinal ranking is sufficient, the procedure is certainly attractive in a problem-solving environment. In our strive toward perfection, this point should not be forgotten (Luce and von Winterfeldt 1994; Tiley 1994).

VIII. MULTICRITERIA DECISION MAKING AND FACILITY LOCATION

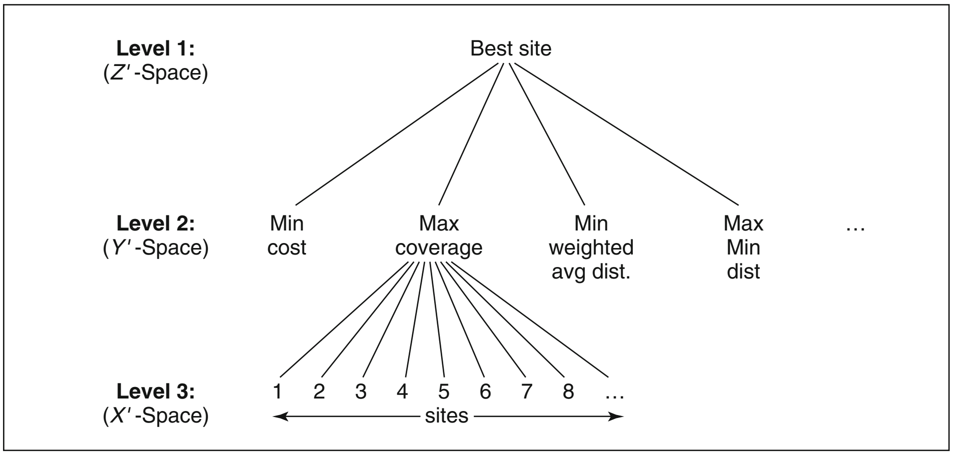
Facility locations are typically evaluated on the basis of multiple criteria. For example, Hegde and Tadikamalla (1990) report on the use of AHP in solving a facility location problem faced by a large multinational corporation. The problem is that of deciding where to locate service terminals for the spare parts division. The AHP was introduced and successfully used to solve the problem. The managers in this case developed a sense of ownership in the findings of the study because the AHP facilitates their involvement at every level. This results in the implementation of the findings from the study. As another example the placement of a landfill has to take into consideration all these factors among others: capital cost, operating cost, environmental impact, and the not-in-my-backyard syndrome (Erkut and Moran 1991). The question is how these criteria can be incorporated into an objective function, if such an exercise is deemed desirable. Here we will demonstrate a way this can be carried out using the MCDM procedures introduced in this chapter.

A. The X, Y′, and Z′ Spaces in Facility Location

In general, three types of objective functions have been used most in the literature concerning location decisions. Covering models locate facilities such that the demands are covered within a pre-specified critical time or distance. Thus express package carriers locate their hubs in such a way in order to capture as much of the market share as possible, while at the same time guaranteeing a specific delivery time. Median or mini-sum models locate facilities in such a way that the average distance between the facilities and the demands served is minimized. An example is the placement of regional distribution warehouses, whereby all the retail outlets are supplied from these warehouses in the most expeditious manner. Center or mini-max models locate facilities to minimize the weighted maximum distance from the facilities to the demands. For example, in locating fire stations, a meaningful criterion is to be able to take care of the fire furthest away from the station as rapidly as possible. Chan (2005) discussed these objectives in detail in his "Facility Location" chapter.

A number of MCDM techniques have been reviewed in this chapter to locate facilities, including MCO, interactive programming, compromise programming, multi-attribute utility theory (MAUT), and AHP. Many of these modeling approaches will again be discussed more substantively in sequel. For the time being, we illustrate the usefulness of AHP, MAUT, and MCO in viewing the role of MCDM in location modeling. The MCDM location model can be structured as shown in the typical AHP tree of Figure 5.22, which is seen to be totally consistent with the X, Y' and Z' spaces advocated by the author for viewing MCDM. MCO requires first of all an explicit definition of the alternative space (X) and the outcome space (Y'), in order to define the efficient frontier. Where desirable, it may further require the objective function (Z') to be identified before an optimal solution can be found. Interactive programming relaxes the last requirement in that the decision maker can progressively articulate preferences as he or she explores the efficient frontier. MAUT, on the other hand, requires a two-step process of first quantifying the utility function, from which alternatives can then be rank-ordered according to the "common currency of exchange," utiles. Finally, AHP is a self-contained procedure to rank-order alternatives.

Figure 5.22 LEVELS OF HIERARCHY FOR THE LOCATION PROBLEM



SOURCE: Haghani (1991). Reprinted with permission.

We have shown that an MCDM problem can be decomposed into the X, Y' and Z' levels. The relationship between the alternative space (X) and the outcome space (Y') or the objective space (Z') is relatively straightforward, in as much as it amounts to a bookkeeping process once data become available. However, it is much more challenging to provide the relationship between levels Y' and Z' (Beroggi and Wallace 1995). We would like to illustrate this interaction via a multi-criteria optimization case study of locating fire stations (Mirchandani and Reilly 1987). We will show how MAUT can reduce a MCDM problem into a single objective optimization problem by way of utiles. From there on, the problem can be solved as a median (or center) problem, defined between the X space and the Z' space. To obtain a direct relationship between the response time of fire units (the Y' space) and the property or casualty damage (the Z' space) is quite difficult for several reasons, most of which have to do with data availability. Problems also exist owing to uncertainty about when and where a fire might occur, and how DMs value the levels of achievement of various performance measures. Even if it were known how location decisions influence the level of achievement of these performance measures, subjective assessment of the relative values of the attributes associated with these measures would have to be made. An alternative to obtaining an empirically derived cost benefit function is to use utility analysis. This method uses the experience of fire fighting professionals to make the tradeoffs of the various attributes, incorporating uncertainties in the exact future location, number of fires, and response times.

B. Multi-Attribute Utility and Optimization

Let us temporarily assume that the arrival time of the fire trucks fully determines all adverse consequences of fires. Thus a fire truck arriving promptly on the scene results in the least property damage and casualties, and a late arrival results in the worst damage and casualties. Each siting plan yields a probability density function (PDF) for the fire unit's response time, where the PDF takes into account the uncertainties regarding where a fire might occur. Rather than minimizing the expected value of its response time, utility theory suggests that the DM maximizes some function of the distribution of response time, called the expected utility, which is defined as $E[v(\tau)] = \int_0^\infty P_i(\tau)v(\tau)\,d\tau$ according to Equation 5.10 where $P(\tau)$ is the PDF of the response time to a random fire, and $v(\tau)$ is the utility of response time τ. Notice the utility of response time needs not be a univariate linear function. For example, a fire chief might prefer a 3-minute expected response time with a low variance to a to 2-minute response time with a high variance.

If we partition the study area into n' zones, the expected utility can be represented as

$$E[v(\tau)] = \sum_{i=1}^{n'} f_i \left[\int_0^\infty P_i(\tau)v(\tau)\,d\tau \right]$$

where f_i is the proportion of fires in zone i, and $\mathrm{P}(\tau)$ is the PDF response time τ to the random fire in zone i. Note that $P_i(\tau)$ depends on the location of the closest fire truck unit to zone i. Thus suitable location criterion for the scenario is to maximize the expected utility, $E[v(\tau)]$, by optimally placing the required p units among the available n' sites.[5] The practice of a typical fire department calls for dispatching a pre-assigned number of units from pre-specified locations to a fire. The philosophy is that more than one fire truck is often required to put out a fire. In this case, we assume that the first two fire trucks dispatched are the most critical to the outcome of a fire. It is likely that the response time of the second arriving unit will have some effect as the first on the damage caused by a fire. We need a multi-dimensional utility function $v(\tau_1, \tau_2)$, where τ_1 is the first unit response time and τ_2 is the second unit response time. The utility resulting from a given set of fire station locations can now be represented as

$$E[v(\tau_1, \tau_2)] = \sum_{i=1}^{n'} f_i \left[\int_0^\infty \int_0^\infty P_i(\tau_i, \tau_2)\, v(\tau_i, \tau_2)\, d\tau_1\, d\tau_2 \right] \tag{5.16}$$

where $P_i(\tau_1, \tau_2)$ is the joint PDF of the first- and second-unit response times to a random fire in zone i, and $v(\tau_1, \tau_2)$ is the bivariate utility function of the first and second unit response times.

The Taylor series expansion of $v(\tau_1, \tau_2)$ around $(\bar{\tau}_1, \bar{\tau}_2)$, the mean values of respective response times, is

$$\begin{aligned} v(\tau_1, \tau_2) = {} & v(\bar{\tau}_1, \bar{\tau}_2) + v_{10}(\bar{\tau}_1, \bar{\tau}_2)(\tau_1 - \bar{\tau}_1) + v_{01}(\bar{\tau}_1, \bar{\tau}_2)(\underline{\tau_2} - \bar{\tau}_2) + \\ & \tfrac{1}{2} v_{20}(\bar{\tau}_1, \bar{\tau}_2)(\tau_1 - \bar{\tau}_1)^2 + \tfrac{1}{2} v_{02}(\bar{\tau}_1, \bar{\tau}_2)(\tau_2 - \bar{\tau}_2)^2\, v_{11}(\bar{\tau}_1, \bar{\tau}_2)(\tau_1 - \bar{\tau}_1)(\tau_2 - \bar{\tau}_2) + \cdots \end{aligned} \tag{5.17}$$

where $v_{ij}(\tau_1, \tau_2)$ are the partial derivatives corresponding to the ith and jth order:

$$v_{ij}(\tau_1, \tau_2) = \left(\frac{\partial}{\partial \tau_1}\right)^i \left(\frac{\partial}{\partial \tau_2}\right)^j v(\tau_1, \tau_2)$$

Since $\bar{\tau}_1, \bar{\tau}_2$ are the mean values of τ_1, τ_2 respectively, then $E(\tau_1 - \bar{\tau}_1) = 0$ and $E(\tau_2 - \bar{\tau}_2) = 0$, and the expected value of $v(\tau_1, \tau_2)$ can be approximated by

$$\begin{aligned} & v(\bar{\tau}_1, \bar{\tau}_2) + \tfrac{1}{2}E[v_{20}(\bar{\tau}_1, \bar{\tau}_2)(\tau_1 - \bar{\tau}_1)^2] + \tfrac{1}{2}E[v_{02}(\bar{\tau}_1, \bar{\tau}_2)(\tau_2 - \bar{\tau}_2)^2] \\ & \quad + E[v_{11}(\bar{\tau}_1, \bar{\tau}_2)(\tau_1 - \bar{\tau}_1)(\tau_2 - \bar{\tau}_2)] \end{aligned} \tag{5.18}$$

or equivalently

$$v(\bar{\tau}_1, \bar{\tau}_2) + \frac{1}{2}v_{20}(\bar{\tau}_1, \bar{\tau}_2)\sigma_1^2 + \frac{1}{2}v_{02}(\bar{\tau}_1, \bar{\tau}_2)\sigma_2^2 + v_{11}(\bar{\tau}_1, \bar{\tau}_2)\ \mathrm{cov}\ (\tau_1, \tau_2) \tag{5.19}$$

The above expression gives the expected utility of a siting plan as a function of the means, variances (σ), and the covariances (cov) of the response times of the two first arriving units.

Through a series of structured interviews with an official of the Albany, New York Fire Department, Mirchandani and Reilly (1987) developed a utility function of the response times for the first two engines arriving at a fire. To start out, single-attribute utility functions were assessed. The multi-attribute utility functions were then constructed from these univariate functions. To illustrate, consider the utility function for the first engine response time to low-risk fires. The fire department official revealed that he was constantly risk averse. A generic form to represent risk-averse univariate function is $v(\tau) = a - be^{c\tau}$, where a, b and c are positive calibration constants. During the interview, the official indicated that he would be indifferent between a 50-50 lottery of "1-minute and 5-minute response times" and a "response time of 3.75 minutes with certainty." The utility functions were assessed by conducting such lotteries over a range of 0–10 Min. By assigning the following arbitrary utility values to the two extreme outcomes, $v(0) = 1$ and $v(10) = 0$, the following utility function results for the first and second engines that arrive, respectively,

$$v_1(\tau) = 1.016 - 0.016e^{0.415\tau}$$

and

$$v_2(\tau) = 1.079 - 0.079e^{0.262\tau}$$

By assuming mutual preferential and utility independence, the multi-attribute utility-function $v(\tau_1, \tau_2)$ can be shown as

$$0.971v_1(\tau_1) + 0.763v_2(\tau_2) - 0.734v_1(\tau_1)v_2(\tau_2)$$

Details of the study are documented in Reilly (1983). The discussion above simply illustrates that MAUT can be used in a practical scenario to formulate a single objective function, with which a conventional integer program between the X and Z' space can then be formulated and possibly solved, using a median formulation in this case. Thus the two seemingly disparate bodies of knowledge of MCDM, MAUT and MCO, can in fact be integrated productively into a single package for real world applications. This simple case study amply illustrates emergency facility location problems that are found not only in providing urban services, but also in such sectors as the defense community, where tactical operations are often judged on the basis of timeliness in response.

IX. A TAXONOMY OF METHODS

In the above brief review of MCDM, we regard the DM's underlying problem as one of selecting an alternative y' from the set of alternatives Y' in criteria space so as to best achieve his/her objectives as reflected by the value function, $v(\mathbf{y}')$: $\text{Max}\{v(\mathbf{y}') \,|\, \mathbf{y} \in Y'\}$. To the extent that information is decentralized and not immediately available, an educational process is required on both the part of the DM and the analyst in order to gain insights into the problem. Examples abound in both interactive mathematical programming and multi-attribute value function definition. Table 5.2 shows such an educational process (Bogetoft and Pruzan 1991).

In this table, it is clear that considerable interaction between the DM and the analyst is a prerequisite for a successful MCDM process. Both the DM and the analyst will have to be willing to assume part of the responsibility in either initiating or responding to a particular MCDM procedure. Thus a procedure can be directed by either one of the two parties. The type of interaction can be either one-way or two-way, as shown by the arrows in the table. Obviously, a two-way interaction is by definition more involved than one-way, resulting in an iterative process.

A. Prior Articulation of Alternatives

The analyst can undertake an extensive investigation of the set of feasible alternatives, Y′, and submit them as a set of proposals to the DM. The DM inspects the set of proposals, clarifies his own preferences and makes a choice. Throughout this chapter, we have pointed out simple ordering as a key concept in organizing the alternative set, particularly in defining the efficient frontier. This value free Pareto concept allows some powerful computational procedures to be followed in MCO. While the

Table 5.2 TAXONOMY OF INTERACTIVE MCDM PROCEDURES

Type of investigation	DM directed	Analyst directed
Phased	Prior articulation of alternatives (DM ← Analyst)	Prior articulation of preferences (DM → Analyst)
Iterative	Progressive articulation of alternatives (DM –q→ analyst) ←a–	Progressive articulation of preferences (DM←q– analyst) –a→

weighted sum procedure is readily operational with many off-the-shelf LP software, its application is mainly limited to problems amenable to LP model formulations. The constraint reduced feasible region method is likely to be more versatile in solving integer programming and nonlinear programming models. To the extent facility location decisions are discrete, integer programming is a key technique in the tool box of the analyst and so is the constraint reduced feasible-region method.

B. Prior Articulation of Preferences

In this model, the DM's preferences $v(\mathbf{y}')$ is constructed by the analyst based on studying the DM's behavior through surveys (for example). In this process, the objectives, criteria and attributes of the stakeholders need to be first defined, say, in a multi-attribute value function. The preference structure of the DM is then subsequently established in terms of such verifiable properties as transitivity of preferences and preferential and utility independence among attributes. These properties will point toward a particular way that alternatives can be rank ordered, including ordinal, cardinal, and lexicographic ranking. If transitivity is established, for example, value/utility functions can possibly be calibrated to operationalize the repertoire of techniques under MCO or **multi-attribute decision analysis** (MADA), with the latter defined here as the generalization of single-attribute Bayesian decision with the latter theory.

C. Progressive Articulation of Alternatives

Instead of a one-way interaction, either directed by the DM or the analyst, this process is now iterative. In each iteration, the DM asks the analyst about the set of alternatives, the analyst answers and the DM evaluates the answer. The DM then decides either to continue the search by posing new questions or to stop the search and choose one of the alternatives identified so far. Compromise programming is one of the ways that such iteration can take place. If no satisficing solutions exist, the goal setting process can compromise between the goals and the set of feasible alternatives. The set of alternatives, as reflected through the criterion space, or the Y'-space, can be greatly expanded—for example—upon the availability of additional resources, including monetary, technological, or managerial. This will allow satisficing solutions to be generated to meet the expectations of the goals set forth by the DM.

D. Progressive Articulation of Preferences

In each iteration, the analyst poses questions to the DM about his preferences and the DM answers. If the analyst now knows sufficiently about the DM's preferences to make a choice from the concrete set of alternatives, he/she proposes a choice. Otherwise, the questioning continues. In MCLP, for example, the analyst can assist the DM to "travel along" the efficient frontier, wherein the DM goes back and forth between the relevant adjacent efficient extreme points (See Section IV-C on the interactive Frank-Wolfe algorithm.) Through this process, the analyst and the DM both sharpen their insights into the problem, allowing for a coordinated effort in decision making (Huang and Li 1994). As part of the progressive articulation of preferences, it is entirely possible that the domination structure of the DM falls outside the paradigm of Pareto preference and value function. Both of these concepts are indeed very restrictive, and a large gap exists between the two (as will be explained immediately below). While this will nullify a fair amount of the formal methodology presented above, it nevertheless allows the DM to obtain the most important product of the process: insights into the problem at hand.

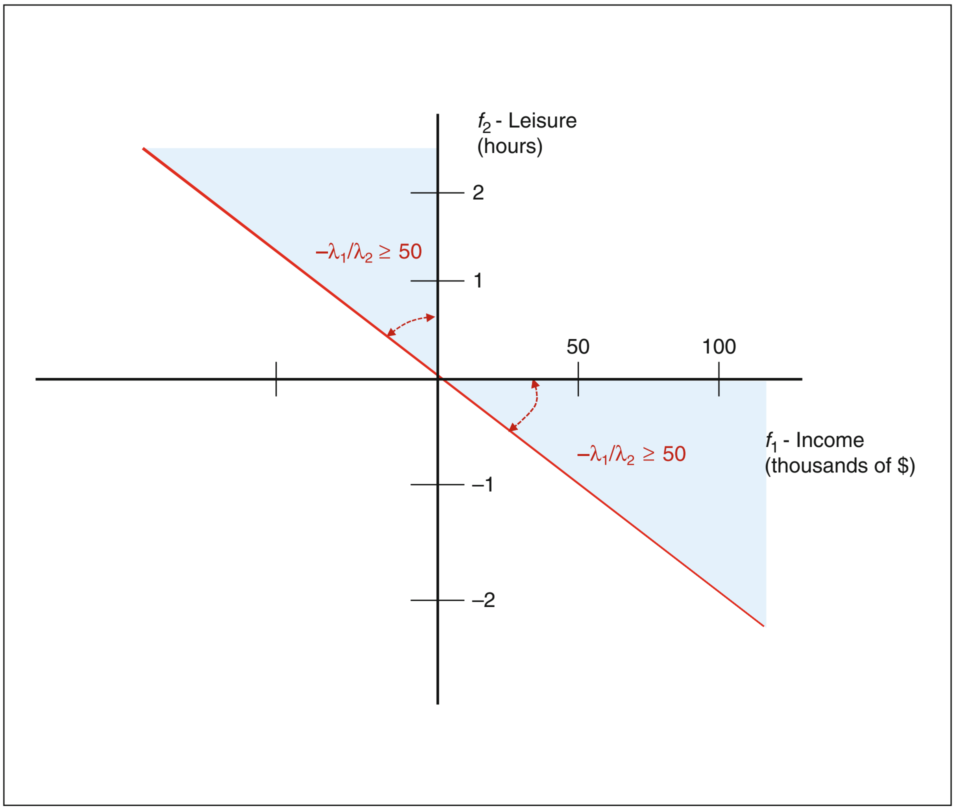
X. DOMINATION STRUCTURES

Pareto preference is the simplest kind of preference, which allows MCO to be operationalized. On the other hand, the assumptions of a value function representation or the existence of preference relationship $\{ > \}$ certainly are very restrictive. The gap between the assumption of Pareto preference and that of the preference having a value function representation is very large (Li and Sinha 2004). Let us give an example (Yu 1985). A landowner is willing to sell the land adjacent to his or her home for at least $50,000/acre (125,000/ha). Let f_1 be the extra leisure time in hours he or she would spend in his homestead should he or she retain the adjacent land and f_2 be the additional income in dollars generated from the land sale. In terms of the weight cone, he or she requires $df_2/df_1 = -\lambda'_1/\lambda'_2 \geq 50$ in $v(\boldsymbol{\lambda}) = \lambda'_1 f_1 + \lambda'_2 f_2$ for a sale to occur. Figure 5.23 shows his or her domination structure. Observe that unless Y' is convex, the non-dominated set cannot be obtained using maximization of the value function $v(\boldsymbol{\lambda}')$. In other words, the final solution may not be obtainable by maximizing additive or multiplicative value functions.

Can we somehow make transformations to the criteria f_1 and f_2 that will allow us to derive a value function that we can maximize? The landowner reveals that he or she is happy to sell for at least $50,000 per acre. The question here is that for $50,000 an acre; is the landowner willing to give up one, two, three, or more acres? What about $60,000 per acre? In this problem, the landowner has only a preference, while a value function attempts to create cardinality among these one-, two-, or three-acre land sales. In this case, he or she is willing to sell for more than 50,000 an acre, but is not sure how much land to sell. For example, at $50,000 per acre, he or she might be willing to sell one acre. At that point the $50,000 might no longer be worth missing any additional acres. However, at $100,000 per acre, he or she might be willing to sell more, because now he or she would be earning enough additional money to further improve and expand on the existing home (even though there is less surrounding land to enjoy.) Questions such as these offer insights into the marginal rate of substitution between leisure hours and extra income. We typically plot a marginal rate of substitution curve as convex function with which a convex Y' region can be applied, and an optimal solution can be identified (See illustration in Figure 5.23.) With the information given, however, it is not clear whether the marginal rate of substitution curve is concave or convex, and as a result, no optimal solution can be obtained.

In summary, the information available from the DM is only sufficient to give a rank order, in other words, $51,000 per acre is preferred to x hours of additional leisure hours at the homestead. Even this is dubious, since we cannot valuate x. We cannot derive a value function because we cannot derive the univariate utility function for leisure time and extra income. In other words, the model fails

Figure 5.23 WEIGHT CONE FOR ALTERNATIVE DOMINATION STRUCTURE



f_2 - Leisure (hours)
2
$-\lambda_1/\lambda_2 \geq 50$
1
50
100
$-\lambda_1/\lambda_2 \geq 50$
f_1 - Income (thousands of $)
–1
–2

because the revealed preference does not predict the DM's answer if we ask a question such as: "Would you sell *five* acres for $50,000 an acre?," which by the way is a legitimate question for the buyer to ask.

XI. COLLECTIVE DECISION MAKING

Following the same line of reasoning, the design that is best overall for a group of individuals with different interests often cannot be found analytically either. There cannot be any universally acceptable analytic solution because people place enormously different values on products. The utility/value function for a group is known formally as a group utility/value-function (GUF), which is an aggregation of individual values GUF = $f(v^1, \ldots, v^n)$. How does the aggregation of individual value functions into GUF take place? Should it be simply the weighted sum, GUF = $\Sigma_i w^i v^i$? But how about equity among the individuals that make up the group—that every one should achieve some minimal level of satisfaction, and we should discount a disproportionately excessive individual level of utility? Perhaps GUF = $\Sigma_i w^i \exp(-[v^i - E(v)])$ may be more appropriate? This function is maximized when $v_i = \bar{v}$.[6] Keeney and Kirkwood (1975) define a **group utility function** (GUF) as an extension of the multi-attribute utility function specified in Equation 5.11. The additional weights w^i represent value tradeoffs of the decision makers. According to the authors, the GUF can be specified by a dictator who picks weights impartially to incorporate the preferences of all group members into the decision, or by using the collective response of the group to define the weights. In the first case, the process is similar to the technique used to determine parameters for a single DM. The second case involves a combination of the individual's utility functions and evaluation of the **individual group utility function** (IGUF) for each member of the group. The GUF is then constructed as a weighted aggregation of the IGUFs. This process includes interpersonal comparison of preferences and requires the measurement of the strength of individual preferences. Given complexities associate with the above method, it can be difficult to determine the overall GUF.

A. Arrow's Paradox

The difficulty of group decision making is well-publicized by the Arrow's paradox (Arrow 1963), which highlights these salient points:

1. The choices a group makes depend on its internal rules of decision-making; for example, its voting rules.
2. No one voting rule or decision-making process is intrinsically best.
3. The choices made by a group are therefore necessarily an ambiguous reflection of its preferences, so that we cannot rely on a group's choices to construct its GUF.

A voting procedure illustrates ambiguity of choice. Consider a family of three persons evaluating three different houses to buy, with their individual assessments looking like the following (de Neufville 1990).

Following our discussion in Section I in the current chapter, suppose the family agrees to select its home by successively comparing pairs of options until it has ranked them all. Thus comparing A against B, home A is preferred to home B ($A > B$) by a 2:1 majority according to the tabulation above. If home A is then retained as a preferred option and compared to home C, we find $C > A$ also by a 2:1 majority. Having compared all three options, can we conclude $C > A > B$?

To answer this question, we can check the results by comparing B and C, wherein $B > C$ by a 2:1 majority. Thus one can conclude that $C > A > B > C$, which is an intransitive result! This example shows that we cannot rely on the choices expressed by a group to reflect its GUF. The actual choice may depend critically on the precise way a voting or consensus building procedure is applied. This again einforces the conclusion reached from another location decision example documented in Section I-A of the current chapter. Research into this difficult problem is continuing, as evidenced by Leitmann (1976).

Ranking of housing		Family member	
locations	Husband	Wife	Dependent
First	A	C	B
Second	B	A	C
Third	C	B	A

B. Game Theory

A group decision-making process is sometimes modeled by game theory, which tries to capture the pluralistic decision-making process (Silberberg 1990). A historic game-theoretic model of interaction between market participants is Cournot's analysis of a duopoly, or a market in which exactly two suppliers produce identical goods or services. Let V_i^s be the output of firm i ($i = 1, 2$), let $C_i(V_i^s)$ be that firm's cost function, and let $D(V^d)$ represent the industry inverse-demand curve, or the price-schedule expressed as a function of the firms' total output V^d, where $V^d = V_1^s + V_2^s$. A downward sloping demand function is assumed. If the firms were able to collude perfectly, that is, act as monopolist, they could achieve maximum profit since between the two of them they have cornered the market. Cournot considered the case where such collusion was impossible. He postulated that, at any moment, each firm would maximize its profits assuming the other firm's output as given. In other words, a firm's decision is based on the other firm's output decision from (yesterday), fully convinced that this other firm will behave the same manner (tomorrow). Each firm maximizes its profits based on this somewhat naive assumption. In this way, the firms continually adjust their outputs until each firm has no further incentive to do so.

Take the example of the demand curve $D(V^d) = 30 - V^d = 30 - V_1^s - V_2^s$ and the constant-cost curves $\dot{C}_1(V_1^s) = \dot{C}_2(V_2^s) = 6$. For firm 1 therefore, the profit-maximization objective function would look like $\max_{V_1^s} I_1 = [D(V_1^s + V_2^s)V_1^s - C_1 V_1^s]$ with a similar expression for firm 2. In our example, the profit to be maximized is $I_1 = (24 - V_1^s - V_2^s)V_1^s$. When the other firm's output is taken as parametric, the first-order conditions for optimization are obtained simply by the partial derivatives

$$V_1^s \dot{D}\,(V_1^s + V_2^s) + D - \dot{C}_1(V_1^s) = 0 \textit{ and } V_2^s \dot{D}(V_1^s + V_2^s) + D + \dot{C}_2(V_2^s) = 0 \text{ respectively}$$

In this example, we are taking the partials as the following:

$$\frac{\partial I_1}{\partial V_1^s} = 24 - 2V_1^s - V_2^s = 0 \text{ and } \frac{\partial I_2}{\partial V_2^s} = 24 - 2V_2^s - V_1^s = 0$$

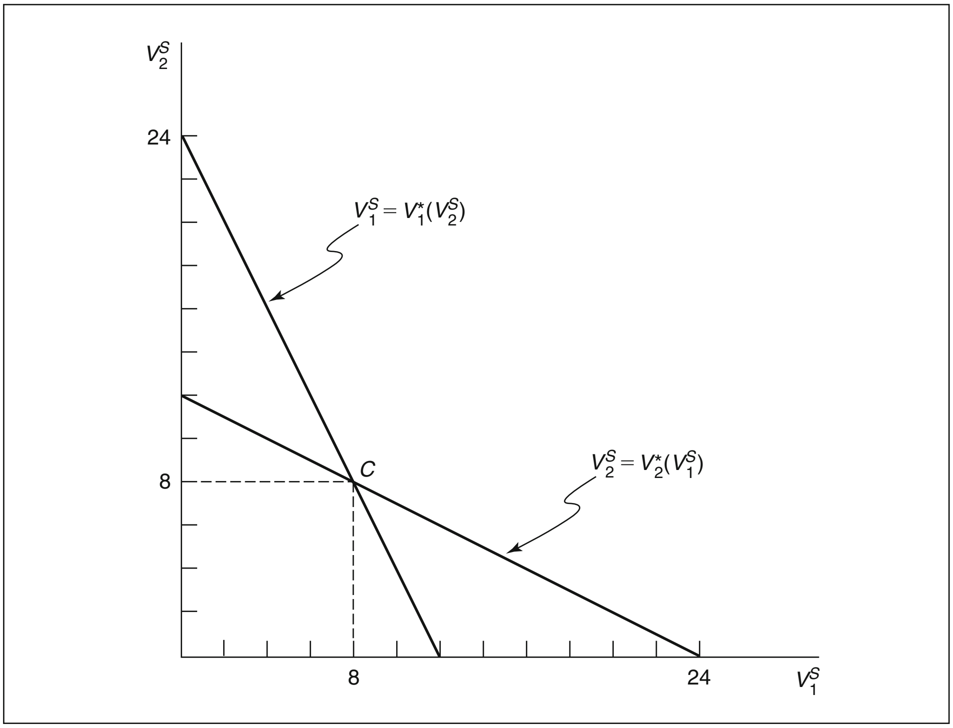
Solving each equation in terms of the other firm's output yields the reaction functions

$$V_1^s = V_1^*(V_2^s) \quad V_2^s = V_2^*(V_1^s) \qquad (5.20)$$

For this example, the reaction functions are $V_i^s = 12 - V_j^s/2$ ($i = 1, 2$), as illustrated in Figure 5.24. For the general case where the demand function is nonlinear, the reaction functions are shown in Figure 5.25 as curvilinear curves. Both of these reaction curves, whether linear or curvilinear, indicate the profit maximizing output of each firm, for parametric values of the other firm's output. The intersection of the firm-1 and firm-2 curves, point C, is the Cournot solution to the duopoly market. In our example, $V_1^s = V_2^s = 8$, or the firms will provide 8 units of output each, at a market price of 14 units, and yielding a net profit of 64 units.

This solution is also called a **Nash equilibrium,** defined as the more general equilibrium condition in which neither firm will change its decision

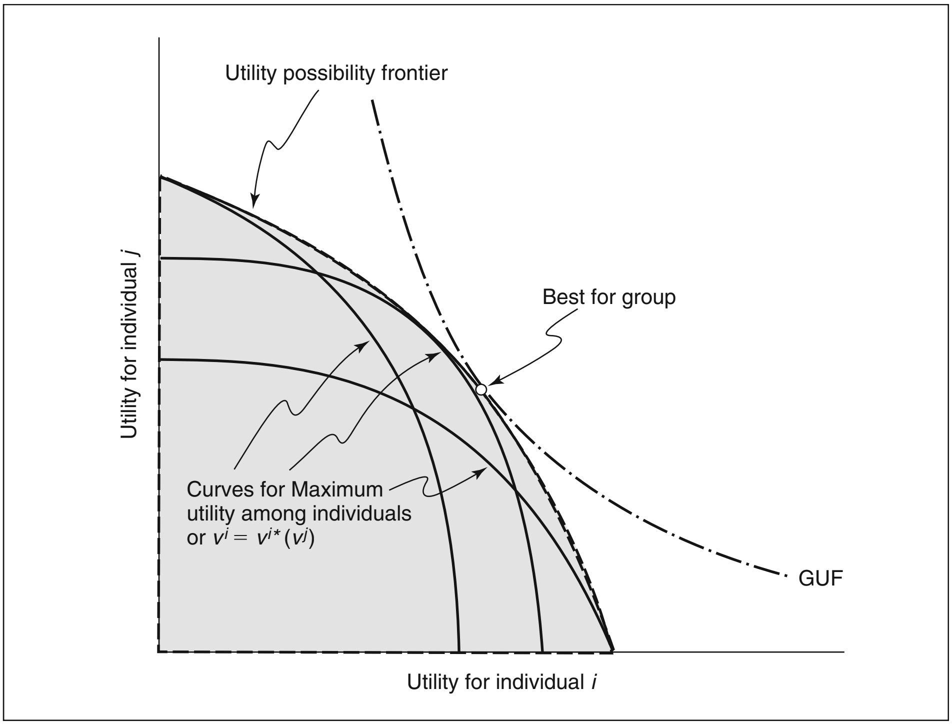
Figure 5.24 NASH EQUILIBRIUM IN A COURNOT DUOPOLY



under the assumed behavior. However, it is not a Pareto solution, since further gains from trade could exist with a lower price. For that reason, each firm has a wealth-maximizing incentive to cheat on this arrangement. It can be seen from the demand and profit functions of this numerical example that firm 1 can raise its profit I_1 by lowering it price $D(V^d)$ and increasing its production. Thus by lowering its price (from 14) to 13, for example, firm 1 can capture the entire market and now increase its profit from 64 to (24 – 17) 17 = (7)(17) = 119 units. For this reason, there are variants to this model. Bertrand and Edgeworth, for example, constructed a similar model in which price, instead of output, is assumed fixed in the duopolistic competition. Stackelberg proposes that one firm, say firm 2, assumes that firm 1 will react in a Cournot manner to firm 2's output decision, according to the reaction function (Eqation 5.20). Firm 2 then chooses output assuming the above relation for V_1^s. In other words, it does not assume, as in the Cournot model, that firm-1's output will be fixed. Rather, firm-2 anticipates firm-1's Cournot behavior. Firm 2 in this case is the Stackelberg leader; firm 1 is the follower. Depending on the cost functions, different solutions emerge. Both firms could choose to be leaders, in which case Stackelberg warfare results.

In general, game theory describes the complex behavior of these DMs in a pluralistic setting. It can be shown that game theory, a complicated body of knowledge in and of itself, can become overwhelmingly complex by the introduction of multiple criteria. Building on the work of Cook (1976), Hannan (1982), and Zelany (1976), Patterson, Horton, and Chan (1994) and

Figure 5.25 UTILITY PRODUCTION FRONTIER AND THE GUF


Utility possibility frontier
Best for group
Utility for individual *j*
Curves for Maximum utility among individuals or $v^i = v^{i*}(v^j)$
GUF
Utility for individual *i*

Payne (1995) experimented with the simple case of a two-person zero-sum game, the solution of which corresponds to the primal and dual solutions of an LP assuming maximum gain for one team and minimum loss for the other. The term zero sum here refers to the condition that one team's gain is exactly equal to the other team's loss. As soon as two criteria or two payoff metrics are involved there appeared to be more than one equilibrium, considering both local and global optima. Finding these equilibria is often a trial-and-error process. By now, one should be totally convinced that the analysis of pluralistic decision making is in fact beyond the state-of-the-art in modeling.

C. Recommended Procedure

Since there is no analytic way for a group to choose among options, we can only recommend a procedure for dealing with the problem (de Neufville 1990). The purpose of the procedure then would assist in the meeting of the minds:

1. Model the physical alternatives
2. Define the noninferior options, or sometimes referred to as the production possibility frontier, or trace out the reaction functions
3. Determine individual preferences
4. Explore the possible tradeoffs
5. Negotiate toward a collectively satisfactory solution.

Step 4 unveils the differences between individual's tradeoffs among attributes, thus offering the possibility of mutually beneficial exchanges. In Step 5, the negotiation takes place in the consequences of the alternatives and their cost benefit distribution among the stakeholders.

A compromise alternative in the production possibility frontier may then be identified. A mutually beneficial distribution may result in sharing, instead of monopolizing, the cost-benefits. Figure 5.25 illustrates such a possibility, where the best alternative for a group involves finding both the best alternative and the best way to allocate its cost-benefits. Thus a production possibility frontier is constructed as an envelope of maximum utility curves among all pairs of individuals i and j. The best alternative is then defined as the one that satisfies both the GUF and the production possibility frontier—if a GUF can be defined of course. In lieu of a production possibility frontier, one can think of a triopoly instead of a duopoly and progressively toward an oligopoly market, which collectively define a utility possibility frontier based on individual maximum utility reaction functions $v^i = v^{i*}(v^j)$.

Lewis and Butler (1993) described and evaluated an iterative technique to facilitate multi-objective decision making by multiple DMs. The proposed method augments an interactive MCO procedure with preference ranking tool and a consensus-ranking heuristic. Computational experience suggests that the proposed framework is an effective decision-making tool. The procedure quickly located excellent compromise solutions in a series of test problems with hypothetical DMs. In addition, a real-world resource allocation study yielded positive feedback from the participants.

XII. CONCLUDING REMARKS

The relative newness of multiple-criteria decision making (MCDM) brings with it a host of competing approaches. The purpose of this chapter is to expose the reader to a wide variety of techniques. The discussion is organized around the paradigm of the X, Y' and Z' space, which allows us to introduce the concepts of decisions, criteria, and value functions systematically, and cover both multple-attribute decision analysis and multiple-criteria optimization (MCO). It has been shown that Pareto preference is the simplest kind of option ranking requiring little articulation of the decision-makers' preference structure. There is a large gap between this and ranking based on a value function, where the DM's articulation of preference has to be clearly understood. Understanding the DM's value function has to be the most interesting and challenging aspect of the field of MCDM. Once the X, Y' and Z' spaces have been defined, the process of MCDM can be carried out. Recent research points toward an interactive procedure to do this, which represents a prominent direction the field is moving.

What are the challenges facing MCDM in the foreseeable future? Zionts (1992) suggests viewing MCDM to be made up of four different subareas: multi-criteria mathematical programming, multi-criteria discrete alternatives (including integer MCO), multi-attribute utility theory, and negotiation theory. On the macro level, he sees negotiation as a fruitful area of research, since there is but a dearth of understanding in this important subject presently. Before and even after rigorous theories are established, approximations to the theory, possibly along the line of an expert system, may be very useful. He cited the example of MAUT, where additive utility functions are often calibrated in practice even though preferential independence cannot be rigorously proven. An expert system is helpful in helping negotiators understand and structure their own positions, even though a theory of negotiation is far from complete. On a micro level, Zionts suggests examining these topics: A Tshebycheff-, or l_∞-norm, is a good proxy utility function since it can generate all non-dominated solution points. More precisely, minimizing the norm as a quasi-convex function can generate any non-dominated solution points (Gardiner and Steuer 1995). Cone dominance can also be exploited by using nonexistent or dummy solutions, such as an ideal, for comparison purposes. The advantage is to increase the information learned as a result of asking preference questions of DMs. A visual display, possibly through computer graphics, will greatly assist the DMs in performing MCDM analysis. We have seen such an example in Figure 5.10.

Dyer et al. (1992) also provide some collective thoughts on the future of MCDM. Utility functions that go beyond the additive format as explained in this chapter are judged to be worthy of further investigation (Fishburn; 1988; Wakker 1989). Abbas (2009) advanced a multi-attribute utility copula that expresses any continuous, bounded multi-attribute utility function that is nondecreasing with each of its arguments. In terms of single-attribute utility assessments, the function is supposed to be strictly increasing with each argument for at least one reference value of the complement attributes. Under these conditions, the formulation provides a wealth of new functional forms that can be used to model preferences over utility-*dependent* attributes. It also enables sensitivity analyses to some of the widely used functional forms of utility independence. On a parallel vein, there is a need for eclectic approaches that synthesize the meritorious among existing theories and practices to improve MCDM procedures. Interactive MCO, an important area of research identified previously, needs consolidation so that procedure switching can take place as the decision process progresses (Buchanan 1994). To this list we would like to add integer MCO, which is an important application area not supported by adequate computational algorithms (Narula and Vassilev 1994).

Korhonen (1992) outlines his observation regarding recent developmental trends. Recent techniques tend to have these features in common:

1. They do not cramp the DM's style and involvement;
2. They have interactive feedback mechanisms including graphics (El-Mahgary and Lahdelma 1995; Antunes and Climaco 1994);
3. They have a built-in evolutionary process, allowing modification of the model as the analysis progresses;
4. DMs are provided with fast turnaround analysis techniques.

Based on this prognosis, it is clear that MCDM (or sometimes called multi-criteria decision aid [Vincke 1992]) is a most needed analysis technique. Because of its developmental nature, however, it also represents an active area of further research.

XIII. EXERCISES

Self-Instructional Module: RISK ASSESSMENT

(to be found on the attached CD/DVD)[7]

In this text, we are interested in project evaluation, particularly in the assessment of a technology-based option, whether it be airport location or infrastructure improvement. We call it technology assessment for short. Depending on whom you ask, technology assessment and its associated techniques have characteristics ranging from the mystiques of an art to the exact calculations of esoteric mathematical techniques. This activity module will help the reader to develop an assessment method that will complement the concepts introduced in Chapters 2 and 5 of the textbook, entitled "Economic methods of analysis" and "Multicriteria Decision Making" respectively. In Chapter 2, we first discussed Cost-Benefit Analysis. In Chapter 5, we formally discuss evaluation methods based on multiple criteria, going beyond a single, aggregate metric such as cost or benefit.

Technology or project assessment is basically a two-step procedure. The *first* step involves the determination of the short-term effects such as costs and benefits (costs as measured by implementation and design efforts, and benefits as measured by efficiency, productivity, etc.) The *second* step involves the determination of the long-term effects, sometimes called secondary or higher-order effects, on the socioeconomic system.

It is this second step that presents the exceedingly difficult tasks of prediction and anticipation. For example, consider the case of the aerosol-spray-paint cans. When first introduced, their cleanliness and ease-of-use were immediately recognized as benefits. But who would have predicted these same spray cans would in the long run responsible for an increase in cost—witness defaced properties such as New York City's subway trains and stations.

Do not be misled by the already stated two-step procedure of project or technology assessment. It is more gray than black-and-white, more nascent than mature, and sometimes more ad hoc than codified. At the end, however, a decision has to be made regarding the most desirable option or options to follow. This module will begin exploring some of the analytical techniques in such decision-making. This module is divided into two sections. The first section deals with the "nuts and bolts." The second section allows the reader to apply these "nuts and bolts" in several illustrative exercises, ending with an interesting risk-assessment case study.

By the end of this exercise, the reader will have been exposed to:

(a) Examples of rare events with high-value consequences
(b) Risk analysis using event or decision trees
(c) Examples of real-world decision-making.

This module serves as an excellent introduction to the current "Multicriteria Decision Making" chapter, in which risk assessment is an important component. Risk was first introduced in Chapter 3, where Bayesian analysis built upon subjective probability was discussed. The single-metric decision tree is expanded later to include multiple metrics in the current chapter. As with other modules, the reader is motivated by hands-on engagements in the present module, including real-world case studies that the reader can relate to in their daily lives. For the interested readers, more practical examples can be found in Koller (2005).

Problem 1: Multicriteria Optimization

To model realistically, one often has to combine facility location with routing. Furthermore, one has to address the stochastic nature of demand.

(a) Referring to Figure 5.26 as an example, please write out the mathematical- programming formulation for the bicriteria of

- □ Maximizing demand coverage in the routing from starting node 1 to terminus node 4, with different nodal demand for each state *k*; and
- □ Minimizing the travel time on this routing.

In your formulation, please use the binary variable x_{ijk}, to denote whether the demand at node *i* assigned to a facility at node *j* in system state *k*. Also, let the binary variable y_i denote whether a facility is located at node *l*.

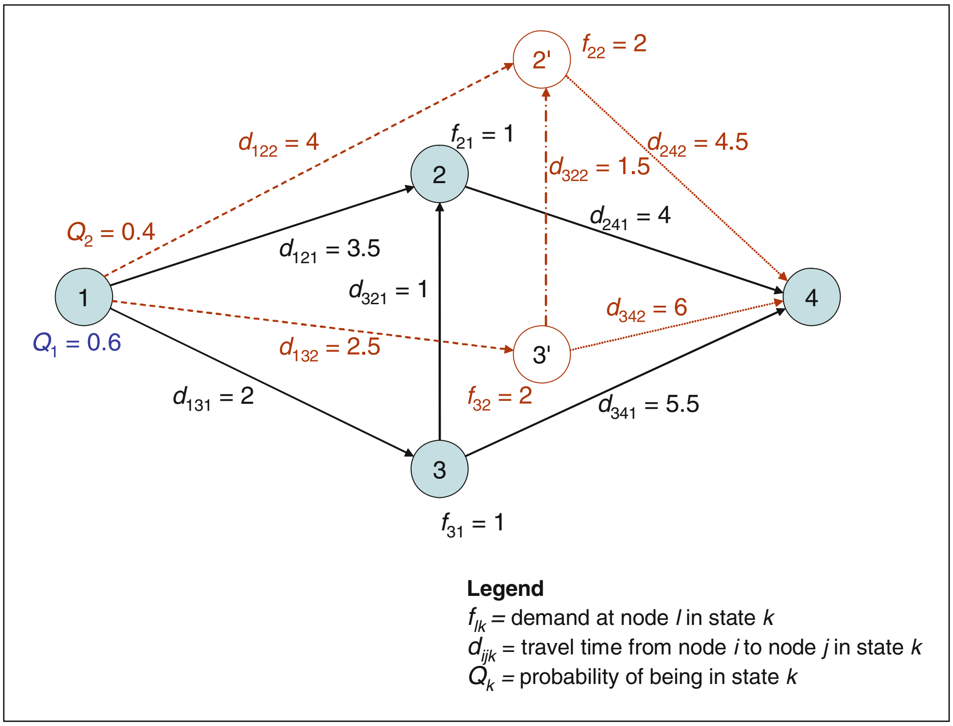
(b) Consider using solution software such as "LINGO" or linear programming software such as *ADBASE*. Generate the nondominated solutions to this mathematical program. (Notice the ADBASE multiple objective linear programming package, obtainable free of charge from Professor Ralph Steuer, at rsteuer@uga.edu.)

(c) Can you guess at the solution of this mathematical program by inspection?

Problem 2: Multi-attribute Decision Analysis

The Metropolitan Planning Organization (MPO) evaluated five conceptual plans for their 2,100 city. All five plans have basically the same cost. Therefore, MPO wants to select the most effective plan

Figure 5.26 STOCHASTIC FACILITY LOCATION AND ROUTING



MPO believes there are only three attributes that will determine the effectiveness of the conceptual plans: (1) Noise foot print (N) of the airport, (2) sustainability index (S), and (3) economic growth (E) potential. Each has the following ranges:

□	$50 \le N \le 100$	Square miles (less is preferred)
□	$0 \le S \le 10$	10-point scale (more is preferred)
□	$400 \le E \le 500$	New jobs/month (more is preferred).

Hired as an expert from Par Excellence University, Dr. Bake decides to construct a utility function, $v(N, S, E)$, using a typical citizen as his decisionmaker. Mr. Doe seems rational (sort-of) and is chosen for the experiment. From Mr. Doe, Dr. Bake solicits the following information.

Set 1: Lotteries over N

$0.5(100) \otimes 0.5(50) \sim 70$	Given:
$0.5(70) \otimes 0.5(50) \sim 55$	$S = 0, E = 400$
$0.5(100) \otimes 0.5(70) \sim 80$	$S = 10, E = 500$

Set 2: Lotteries over S

$0.5(0) \otimes 0.5(10) \sim 4$	Given:
$0.5(4) \otimes 0.5(10) \sim 6$	$N = 100, E = 400$
$0.5(0) \otimes 0.5(4) \sim 2$	$N = 50, E = 500$

Set 3: Lotteries over E

$0.5(400) \otimes 0.5(500) \sim 460$	Given:
$0.5(400) \otimes 0.5(460) \sim 440$	$N = 0, S = 100$
$0.5(460) \otimes 0.5(500) \sim 480$	$N = 10, S = 50$

Set 4: $(50, 0, 400) > (100, 10, 400) \sim (100, 0, 500)$

Set 5: $(75, 0, 400) \sim (100, 10, 400)$

Set 6: $(50, 0, 400) \sim (0.6(50, 10, 500) \otimes 0.4(100, 0, 400))$

Assume that it has been determined that Mr. Doe has a multiplicative utility function. Follow Example 3 of book Section 5-VII-B and answer the following questions:

(a) Draw each of the single attribute utility functions on graph paper.
(b) Determine the total utility function.
(c) Evaluate the expected utility of each of the following proposed conceptual.plans:
(d) What is the best plan? What is the worst plan?

Alternative plans	State θ_1	State θ_2
A	(55, 2, 480)	(70, 4, 420)
B	(70, 4, 440)	(90, 6, 410)
C	(80, 6, 400)	(100, 10, 400)
Prob(State)	$P(\theta_1) = 0.4$	$P(\theta_2) = 0.6$

ENDNOTES

[1] A ranking among alternatives is transitive $A > B$, $B > C$ means $A > C$.

[2] For a review of the simplex procedure, see Appendix 4.

[3] A nondegenerate lottery means that the lottery will have two distinct outcomes, thus discounting the special case where the two outcomes are identical.

[4] Cost is in millions of dollars, time to complete is in years, and effectiveness is the number of shoppers attracted per month (in thousands).

[5] While the p-median problem was introduced in Chapter 4, the “Facility Location” chapter in Chan (2005) discusses the conventional p-median problem in detail.

[6] Notice the exponential form for v^i is simply written for conceptual clarity, the real function should be $1 - \exp(-|v^i - \bar{v}|)$, which means GUF = $\sum_i w^i [1 - \exp(-[v^i - \bar{v}])] = \sum_i w_i - \sum_i w^i \exp(1 - [v^i - \bar{v}]) = 1 - \sum_i w^i \exp(1 - [v^i - \bar{v}])$. This new function is maximized when $|v_i - \bar{v}| \to \infty$.

[7] The answer to this Module is attached at the end of this textbook.

REFERENCES

Abbas, A. E. (2009). “Multiattribute Utility Copulas.” *Operations Research*. 57:1367–1383.

Ang, A; Tang, W. H. (1984). *Probability concepts in engineering planning and design. vol. II: Decision, risk and reliability.* New York: Wiley.

Arrow, K. (1963). *Social choice and individual values.* New Haven, Connecticut: Yale University Press.

Askoy, Y.; Butler, T. W.; Minor, E. D. III (1994). Comparative studies in interactive multiple objective mathematical programming. Working Paper. A. B. Freeman School of Business. Tulane University. New Orleans, Louisiana.

Antunes, C. H.; Climaco, J. (1994). “Decision aid for discrete alternative multiple criteria problems: A visual interactive approach.” *Information and Decision Technologies* 19:185–193.

Balling, R. J.; Taber, J.; Brown, M. R.; Day, K. (1998). Multiobjective urban planning using a genetic algorithm. Working Paper. Department of Civil and Environmental Engineering. Brigham Young University. Provo, Utah.

Bana e Costà, C. A. (1990). *Readings in multiple criteria decision aid.* Berlin, Germany and New York: Springer-Verlag.

Bana e Costà, C. A.; Enslin, L.; Zanella, I. J. (1998). “A real-world MCDA application in cellular telephony systems.” In *Trends in multicriteria decision making,* edited by T.J. Stewart and R. C. van den Honert. Berlin, Germany and New York: Springer-Verlag, 412–423.

Beroggi, E. G.; Wallace, W.A. (1995). “Operational control of the transportation of hazardous material: An assessment of alternative decision models.” *Management Science* 41:1962–1967.

Bogetoft, P.; Ming, C.; Tind, J. (1994).”Price-directive decision making in hierarchical systems with conflicting preferences.” *Journal of Multi-Criteria Decision Analysis* 3:65–82.

Bogetoft, P.; Pruzan, P. (1991). *Planning with multiple criteria: Investigation, communication, choice.* Amsterdam, The Netherlands: North-Holland.

Briggs, T. H.; Kunsch, P.L.; Mareschal. B. (1990). “Nuclear waste management: An application of the Multicriteria PROMETHEE methods.” *European Journal of Operational Research* 44:1–10.

Buchanan, J. T.; Daellenbach, H.G. (1987). “A comparative evaluation of interactive solution methods for multiple objective decision making.” *European Journal of Operational Research* 29:353–359.

Buchanan, J. T. (1994). "An experimental evaluation of interactive MCDM methods and the decision-making process." *Journal of the Operational Research Society* 45:1050–1059.

Chan, Y. (2005). *Location, transport and land use: Modelling spatial-temporal information*. Berlin and New York: Springer.

Chankong, V.; Haimes, Y. Y. (1983). *Multiobjective decision making—Theory and methodology.* Amsterdam, The Netherlands: North-Holland.

Cook, W. D. (1976). "Zero-sum games with multiple goals." *Naval Research Logistics Quarterly* 23, No. 4:615–622.

de Neufville, R. (1990). *Applied systems analysis: Engineering planning and technology management.* New York: McGraw-Hill.

Dyer, J. S.; Fishburn, P. C.; Steuer, R. E.; Wallenius, J.; Zionts, S. (1992). "Multiple criteria decision making: Multiattribute utility theory: The next ten years." *Management Science* 38, No. 5:685–654.

El-Mahgary, S.; Lahdelma, R. (1995). "Data envelopment analysis: Visualizing the results." *European Journal of Operational Research* 85:700–710.

Erkut, E.; Moran, S. R. (1991). "Locating obnoxious facilities in the public sector: An application of the analytic hierarchy process to municipal landfill siting decisions." *Socio-Economic Planning Sciences* 5, No. 2:89–102.

Fishburn, P. C. (1988). *Nonlinear preference and utility theory.* Baltimore, Maryland: Johns Hopkins University Press.

Gardiner, L. R.; Steuer, R. E. (1993). "Unified interactive multiple objective programming." *European Journal of Operational Research* 71:1244–1259.

Gardiner, L. R.; Steuer, R.E. (1995). Range equalization scaling and solution dispersion in the Tchebycheff method: A preliminary study. Working Paper. Department of Management. Auburn University. Alburn, Alabama.

Goicoechea, A.; Hansen, D. R.; Duckstein, L. (1982). *Multiobjective decision analysis with engineering and business applications.* New York: Wiley.

Haghani, A. E. (1991). "Multicriteria decision making in location modeling." *Transportation Research Record* 1328:88–97.

Hannan, E. L. (1982). "Reformulating zero-sum games with multiple goals." *Naval Research Logistic Quarterly* 29, No. 1:113–118.

Hegde, G. G.; Tadikmalla, P. R. (1990). "Site selection for a sure service terminal." *European Journal of Operational Research* 48:77–80.

Huang, Z.; Li, S. (1994). "The role of cardinal utilities in multiple objective programming." *American Journal of Mathematical and Management Sciences* 14:301–325.

Hokkanen, J.; Lahdelma, R.; Salimer, P. (1999). "A multiple criteria decision model for analyzing and choosing among different development patterns for the Helsinki Cargo Harbor." *Socio-Economic Planning Sciences* 33:1–23.

Ignizio, J. P.; Cavalier, T. M. (1994). *Linear programming.* Englewood Cliffs, New Jersey: Prentice Hall.

Islam, R.; Biswal, M. P.; Alam, S. S. (1996). Clusterization of alternatives in analytic hierarchy process. Working Paper. Department of Mathematics. Indian Institute of Technology. Kharagpur, India.

Karaivanova, J.; Korhonen, P.; Narula, S.; Wallenius, J.; Vassiler, V. (1992). A reference direction approach to multiple objective integer linear programming. Working Paper. Institute of Informatics. Sofia, Bulgaria.

Karasakal, E.; Köksalan, M. (2009). “Generating a representative subset of the nondominated frontier in multiple criteria decision making.” *Operations Research* 57, No. 1:187–199.

Keeney, R. L.; and Raiffa, H. (1976). *Decisions with multiple objectives: Preferences and value tradeoffs.* New York: Wiley.

Keeney, R. L.; Kirkwood, C. W. (1975). “Group decision making using cardinal social welfare functions.” *Management Science* 22, No. 4:430–437.

Keeney, R. L. (1994). “Using values in operations research.” *Operations Research* 42:793–813.

Kirkwood, C. (1997). *Structural decision making.* Belmont, California: ITP Duxbury.

Koller, G. (2005). *Risk Assessment and Decision Making in Business and Industry: A Practical Guide*, 2nd ed. Boca Raton, Florida: CRC Press.

Korhonen, P. (1992). “Multiple criteria decision support: The state of research and future directions.” *Computers and Operations Research* 19:305–307.

Korhonen, P.; Laakso, J. (1986). “A visual interactive method for solving the multiple criteria problem.” *European Journal of Operational Research.* 24:277–287.

Lai, Y-J.; Liu, T-Y.; Hwang, C-L. (1994) “TOPSIS for MODM.” *European Journal of Operational Research* 76:486–500.

Leitmann, G. Ed. (1976). *Multicriteria decision making and differential games.* New York: Plenum.

Lewis, H. S.; Butler, T. W. (1993). “An interactive framework for multi-person, multiobjective decisions.” *Decision Sciences* 24:1–22.

Li, Z.; Sinha, K. C. (2004). “Methodology for multicriteria decision making in highway asset management.” *Transportation Research Record,* No. 1885:79–87.

Li, Z. F.; Wang, S. Y. (1994). “Lagrange multipliers and saddle points in multi-objective programming.” *Journal of Optimization Theory and Applications* 83:63–81.

Lieberman, E. R. (1991). *Multi-objective programming in the USSR.* Boston, Massachusetts: Academic Press.

Luce, R. D.; von Winterfeldt, D. (1994) “What common ground exists for descriptive, prescriptive, and normative utility theories?” *Management Science* 40:263–279.

Massam, B. H. (1988). “Multi-criteria decision making MCDM techniques in planning.” Vol. 30, Part I of *Progress in planning,* edited by D. Diamond and J. B. McLoughlin. Oxford, England and New York: Pergamon Press.

Mirchandani, P.; Reilly, J. (1987). “Spatial distribution design for fire-fighting units.” In *Spatial analysis & location-allocation models,* edited by A. Ghosh and G. Rushton. New York: Van Nostrand Reinhold, 186–223.

Moulin, H. (1986). *Game theory for the social sciences,* 2nd and Rev. ed. New York: New York University Press.

Narula, S. C.; Vassilev, V. (1994) “An interactive algorithm for solving multiple objective integer linear programming problems.” *European Journal of Operational Research.* 79:443–450.

Patterson, K.; Horton, K. G.; Chan, Y. (1994) Games with multiple payoffs. Working Paper. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson AFB, Ohio.

Payne, R. (1995). Games with multiple payoffs: A rejoinder. Working Paper. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson AFB, Ohio.

Reilly, J. (1983). Development of a fire station placement model with consideration of multiple arriving units. Doctoral Dissertation. Rensselaer Polytechnic Institute. Troy, New York.

Rinquest, J. L. (1992). *Multiobjective optimization: Behavioral and computational considerations.* Boston, Mass.: Kluwer Academic Publishers.

Roy, B. (1977) "A conceptual framework for a prescriptive theory of 'decision aid'." *TIMS Studies in the Management Science* 3:55–64.

Saaty, T. L. (1980). *The Analytic hierarchy process.* New York: McGraw-Hill.

Saber, H. M.; Ravindran, A. (1996) "A partitioning gradient based algorithm for solving nonlinear goal programming problems." *Computers and Operations Research* 23:141–152.

Sainfort, F.; Deichtmann, J. M. (1996). "Decomposition of utility functions on subsets of product sets." *Operations Research* 44:609–616.

Seo, F.; Sakawa, M. (1988). *Multiple criteria decision analysis in regional planning.* Dordrecht, Holland: Reidel.

Shields, M.; Chan, Y. (1991). The extended ADBASE program. Working Paper. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson AFB, Ohio.

Staats, R.; Chan, Y. (1994). Numerical examples of multiple criteria decision making. Working Paper. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson AFB, Ohio.

Steuer, R. E. (1986). *Multiple criteria optimization: Theory, computation, and application.* New York: Wiley.

Stewart, T. J. (1999). "Evaluation and Refinement of Aspiration Based Methods in MCDM." *European Journal of Operational Research* 113:643–652.

Tiley, J. S. (1994). Solvent substitution methodology using multiattribute utility theory and the analytical hierarchical process. Master's Thesis. AFIT/GEE/ENS/945–3. Air Force Institute of Technology. Wright-Patterson AFB, Ohio.

Van Herwijnen, M.; Janssen, R. (1998). "The use of MCDM to evaluate trade-offs between spatial objectives." In *Trends in multicriteria decision making,* edited by T. J. Stewart and R. C. van den Honert. Berlin, Germany and New York: Springer-Verlag, 303–312.

Van Herwijnen, M.; Rietveld, P. (1999). "Spatial dimensions in multicriteria analysis." In *Spatial multicriteria decision making and analysis,* edited by J.-C. Thill. Brookfield, Vermont: Ashgate, 77–102.

Vincke, P. (1992). *Multicriteria decision aid.* Chichester, England: Wiley.

Wakker, P. P. (1989). *Additive representation of preferences: A new foundation of decision analysis.* Boston: Kluwer Academic Publishers.

White, D. J. (1990). "A bibliography on the applications of mathematical programming multiple-objective methods." *Journal of the Operational Research Society* 41:669–691.

Yu, P. L. (1985). *Multiple criteria decision making.* New York: Plenum Press.

Zelany, M. (1976). "Games with multiple payoffs." *International Journal of Game Theory* 4, No. 4:179–191.

Zelany, M. (1982). *Multiple criteria decision-making,* New York: McGraw-Hill.

Zionts, S. (1992). "Some thoughts on research in multiple criteria decision making." *Computers and Operations Research* 19:308–311.

6

Remote Sensing and Geographic Information Systems

"Give me to learn each secret cause;
Let number's, figures, motion's laws
Reveal before me stand;
These to great Nature's scene apply
and round the Globe, and through the sky
Disclose her working hand."
Mark Akenside

Locational and land use studies rely heavily on the availability of data. While one can argue that data are never complete enough to perform analyses, there is also a tendency to collect too much information (or at least collect irrelevant information). Data collection has been facilitated greatly by remote sensing devices such as satellites and computer-based data organization tools such as geographic information systems. With the technological advances in remote sensing and geographic information systems, the data collection effort can theoretically be streamlined. But they also underline a more urgent need to match data against information requirements, such that the relevant data are collected and that they are in the correct format and in sufficient quantity. In this chapter, we wish to review the data base that is required in facility location and land use, mainly from the angle of matching data with analysis requirements. Also included is the processing of such data to bring out the information in as useful a form as pos-sible for application-oriented purposes.

I. DATA IN SPATIAL-TEMPORAL ANALYSIS

Depending on the type of application, the data to be collected would vary. Table 6.1 shows sample data requirements for performing land use modeling in an urban setting. As one can see, a lot of data need to be gathered. Moreover,

Y. Chan, *Location Theory and Decision Analysis*, 2nd ed., DOI 10.1007/978-3-642-15663-2_6,

Table 6.1 TYPICAL DATA REQUIRED IN URBAN PLANNING APPLICATIONS

Data Items
□ total population by place of residence
□ population by age-sex groups by place of residence
□ population by family size groups by place of residence
□ population by annual family income groups by place of residence
□ population by industry groups by place of residence
□ population by occupational groups by place of residence
□ total labor force by place of residence
□ total employment by place of work
□ employment by industry groups by place of work
□ employment by occupational groups by place of work
□ employment by income groups by place of work
□ total annual retail sales by place of sale
□ annual retail sales by retailing groups by place of sale
□ total value of manufactured products by place of manufacture
□ value of manufactured products by industry groups by place of manufacture
□ total government expenditures by place of agency
□ capital and operating government expenditures
□ government expenditures, capital and operating, by agency
□ total person trips by place of destination
□ total person trips by land-use groups by place of destination
□ total market value of land by small area
□ market value of land by land-use groups by small area
□ total market value of land and buildings by small area
□ market value of land and buildings by structural-type groups by small area
□ total housing units by small area
□ housing units by type of structure by small area
□ housing units by density class by small area
□ housing units by condition of structure by small area
□ housing units by age of structure by small area
□ total floor area by small area
□ floor area by land-use groups by small area
□ land area by land-use groups by small area
□ accessibility to region by small area
□ distance (time or cost) to all parts of the region or to the center of the region by small area

such data often need to be collected consistently over more than one period of time to observe a trend. Generally, obtaining this amount of data is costly. The advances in collection and data processing devices do not diminish this resource requirement, even though the cost per unit of information may be lowered. This apparent contradiction is traceable to the fact that a lot of information is often collected superfluously, either due to the ease with which the collection and processing devices work or the lack of care taken in the conduct of such procedures. Invariably, only a tiny fraction of the information gathered is useful, and the information that is really needed is left out. It is essential therefore to be selective in accordance with what data is really required, as suggested above.

A. Resource Requirement

There were reported price tags associated with collecting each piece of information listed in Table 6.1, and many of them signify much time and effort. It is necessary before data collection to assess the resource at hand and to perform a careful tradeoff analysis between the worth of a piece of information and its cost. There are three general categories of costs in facility location and land use modeling: data assembly, model calibration, and analysis and forecasting. In urban applications, for example, data assembly is the most costly, taking up to 30–50 percent of the study budget. This figure can perhaps be generalized to other applications as well. Much of the data assembly cost is attributable to manpower. Taking all requirements into consideration, the time required to collect data is about 4 to 6 person-months in each urban application. This assumes the availability of public domain data sources such as the census and remote sensing data such as that from LANDSAT. The cost of collecting data will become prohibitive, if such data need to be collected from original sources.

When new technology such as geographic information system (GIS) and remote sensing are introduced, the data collection resource requirement picture can become much more complex. Oftentimes, there is an enormous overhead involved in such an introduction. More often than not, the problem boils down to the need to properly match technology against the problem at hand, and the institution has to have the correct environment to foster change. Even though this appears obvious, case after case can be cited where well-intentioned people got burned in the automation process.

B. Assembly of Data Sources

In the context of this book, there are essentially five different categories of data required. The first is labeled **activity,** which includes population and employment in urban applications for example. The second is **land use,** which is a physical description of the site(s). The third is **transportation,** which addresses the accessibility issue that governs the way population and employment distribute themselves in the study area. Transportation goes well beyond facilities such as roads, rail, and terminals to include travel time, distance, and costs in general. The fourth is **infrastructure,** which includes public utilities and other supporting elements. A final category includes information on the **environment,** which pertains to water quality, air quality, noise, and so forth. The source of the first category of information—population and employment, is typically the census, which is conducted by the Bureau of Census in the Department of Commerce every 10 years in the United States and updated every five years. The employment statistics are tabulated by the Standard Industrial Classification (SIC) code. Population is compiled by census tract, while education information may be collected by school districts, although there are recent trends to put them on a more consistent geographic sub-units, as afforded by the advent of GIS.

The second category information, land use, is traditionally survey data, supplemented by aerial photos. In the United States, the information is often coded according to the Standard Land Use Coding Manual published by the Department of Housing and Urban Development. Part of the land use information is the permissible development densities, which deal with (among other items) the height of buildings that are in certain zones. In urban applications, such information is often encoded in zoning maps available from metropolitan planning agencies. In recent years, we have seen the introduction of satellite imagery that greatly expands the type of land use information that is available. In rural applications, land use refers to anything from landform and soils to ecological and vegetative classifications.

The third category information, transportation, is traditionally encoded in highway networks for urban applications. Standard computer programs are available to extract the necessary travel time information between two points in a study area. Trip frequency is needed, that is, information on what percentage of trips taken are of a particular duration. For example, 30 percent of the trips may be under 10 minutes in duration, 50 percent between 10 and 20 minutes, with the remaining over 20 minutes. Transportation or highway agencies are the best source of such information. In rural applications, interregional commodity flows are often required, representing trading that takes place via air, highway, or waterways.

Fourth, the infrastructure information—sewers, water supply, and power—is usually dispersed among the various political jurisdictions and utility companies. Individual communities, states, and countries are often the custodians of these records. Inasmuch as utility companies are highly regulated in the United States, these public agencies often need to be consulted before utility companies are willing to release information beyond basic factual data.

Finally, environmental information of interest lies in a variety of stakeholders: for instance, industries that pollute and those that do not, governmental agencies that oversee public health and safety, and advocacy citizen groups who are watchdogs for conservation. While site-specific and interest-group-specific data gathering is indispensable, remote sensing has increasingly played a more important role in environmental monitoring in recent years. It provides accessible information irrespective of political jurisdiction.

C. Use and Display of Information

In view of the cost of data collection, a cogent question to ask is: "What is the minimum information set, or the absolutely necessary amount of information out of the comprehensive set, that will allow us to do the analyses?" The main idea is to identify substitutes in case a particular piece of information is not available. For example, work trips can be substitutes for employment, and housing can serve as a proxy for population. A desirable strategy is to have information that is readily observable, such as from satellites, instead of from secondary sources. Remote sensing technology has developed to such an extent now that this strategy has become quite feasible.

It is widely agreed that a key element of a GIS is graphical display. There are specifications as to the way that a display should be presented and used. For example, it should be problem-oriented, and it should provide just the appropriate amount of information for the occasion—no more and no less. Figure 6.1 and Figure 6.2 show some rather interesting three dimensional plots of population information in York, Pennsylvania—a focal case study area in the "Exercises and Problems" appendix and the accompanying CD/DVD. Two graphs are

Figure 6.1 PROXIMAL MAP OF DEVELOPABLE RESIDENTIAL LAND IN YORK, PENNSYLVANIA

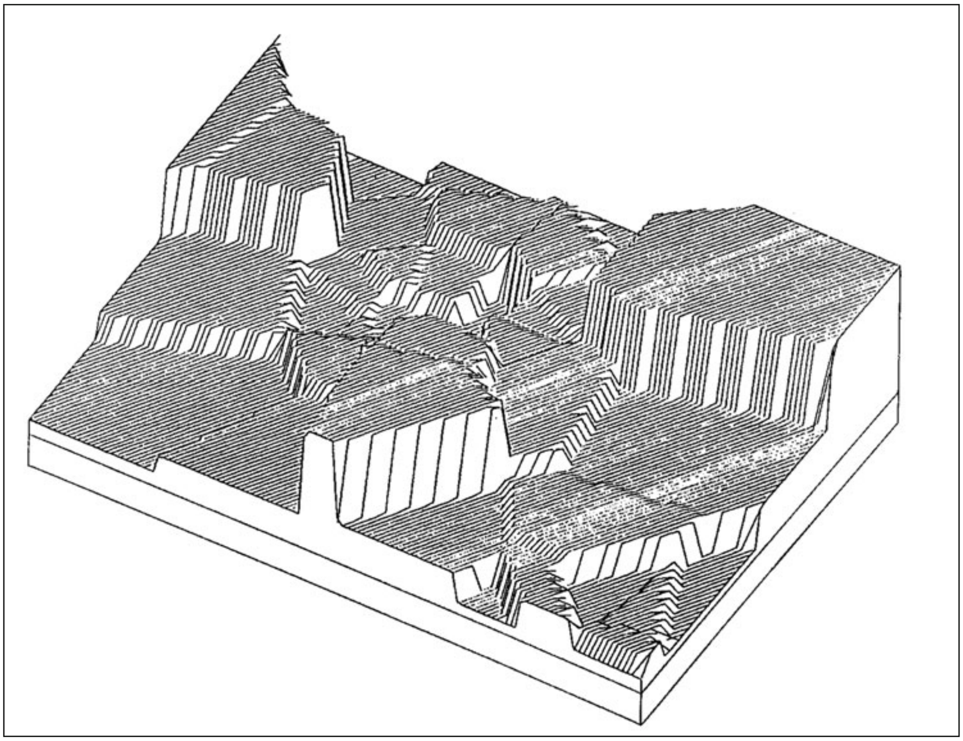
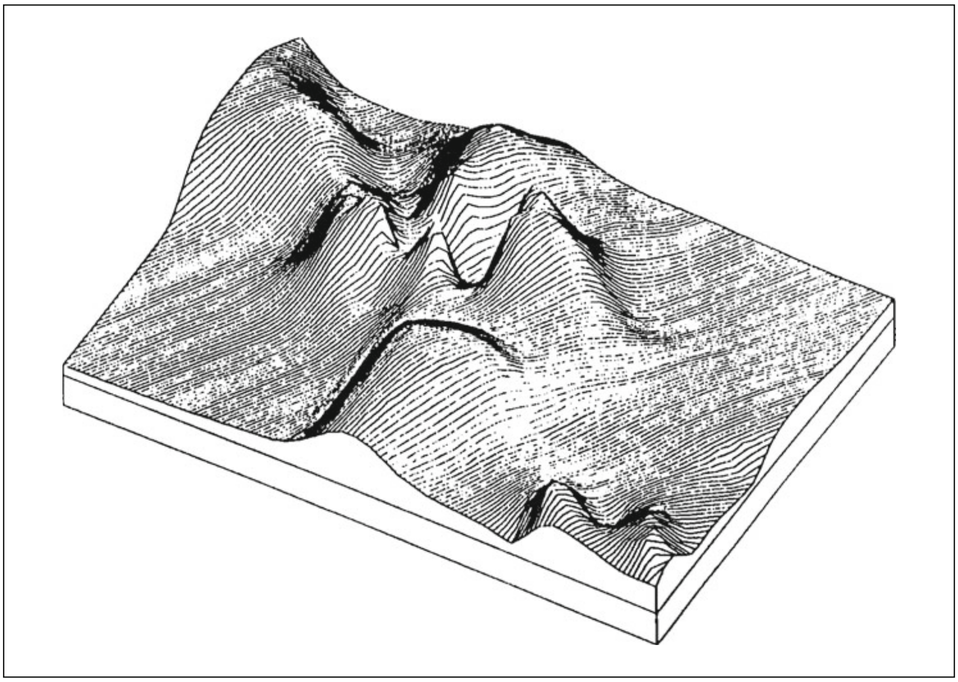


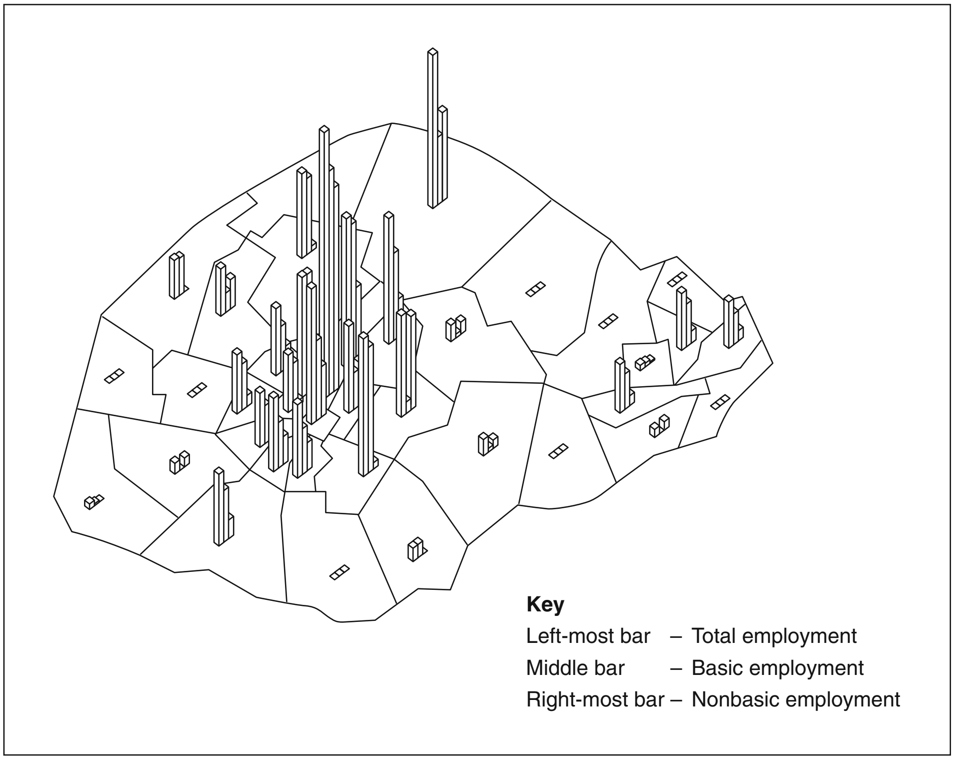
Figure 6.2 CONTOUR MAP OF POPULATION IN YORK, PENNSYLVANIA



displayed—the **proximal graph** by zone (Figure 6.1) and the **contour graph** by continuous distribution (Figure 6.2). Proximal maps are usually graphed for land use information, while contour maps are used for activities such as population and employment. For example, residential land use in thousands of square feet (m^2) can be plotted, delineated by the boundaries of tesselations that approximate traffic zones, as shown in Figure 6.1. Population, on the other hand, is considered to be ubiquitous among developable land and hence represented here as continuous distribution.

Information over time can also be displayed as well (Langran 1992). Zonal population or employment over the base-year and forecast-year can be displayed side-by-side as bar charts in Figure 6.3. Such a plot shows the spatial variation of population or employment activities temporally; it is effective in displaying the regional impacts of a policy over a planning horizon. Obviously, many other variations are possible, including overlays, and there are quite a few graphics packages today that have extensive display capabilities such as virtual reality in which realistic images are constituted by the user for experimentation. Thus existing capabilities in data retrieval and imagery enhancement allow a great deal of flexibility in information display. Perhaps an area for further improvement and exploitation may be a concerted effort to bring the user and the analysis communities together through these graphical displays, so that the analyst can provide the user with what is really needed rather than what the analyst thinks is needed.

Figure 6.3 BASE-YEAR ZONAL EMPLOYMENT, YORK, PENNSYLVANIA


Key
Left-most bar – Total employment
Middle bar – Basic employment
Right-most bar – Nonbasic employment

II. GEOGRAPHIC CODING SYSTEMS

According to Werner (1974), the two major structural elements of all geographic coding systems are a concept of areal division, classification, or definition; and some form of coding logic. In recent years, overt emphasis has been given to the automated aspects of geocoding logic and data storage, retrieval and display in large, geographically referenced information systems. This has resulted in a popular tendency to assume that a legitimate geocoding system must be computer-based and requires a fairly sophisticated coding structure. Broadly conceived, systematic geographic coding has had a long and diverse history that includes many classifications other than coding-oriented systems of geographic reference.

A. Central Place Theory

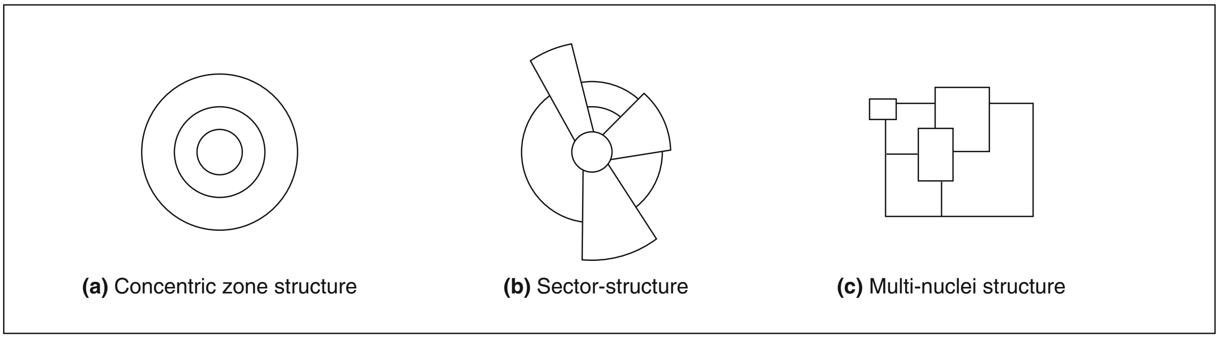
Historically, there exists a well-publicized scheme regarding a natural geographic classification hierarchy. It was believed that central places were developed from distribution points for goods and services in order to serve a surrounding hinterland (for instance, an agricultural region). The central place evolves later on as a political and social center for the region, serving a diverse number of interest groups and concerns beyond farmers. A central place may be developed from a transport focus or break bulk point. For example, Chicago, besides being a central place for the distribution of agricultural goods, is also a natural waterways center and a rail hub for manufacturing industries. It is a distribution and collection center for all commodities passing through the Great Lakes and the Midwest of the United States in general. Specialized function settlements constitute yet another type of central places: for example, coal mining in Scranton/Wilkes-Barre, Pennsylvania. There are many parallel cases of this kind, including resorts, spas, and other natural resource centers.

One can identify a hierarchy of central places. A hamlet is a local center, a village is a neighborhood center; town is a community center; a city is a regional center; and a metropolis and a megopolis may be described as cosmopolitan gathering places. Industrialized nations seem to become more and more urbanized. For this reason, this hierarchical geographic classification scheme may be applicable to a number of industrialized nations. It forms a logical scheme for storing geographic information. Thus one can look up the population and employment in a hamlet versus a village versus a town and all the way up to a megalopolis. Recent analysis techniques organize spatial data around tile-like tessellations that approximate these natural settlement patterns.[1]

B. Concentric Zone, Sector, and Multi-Nuclei City Structures

Aside from these broad classifications of central places, there are some observed regularities in the internal structure of an urban area, upon which finer geographic subdivisions can be discerned. As far back as 1923, Burgess postulated a structure of **concentric rings** around the central business district corresponding to belts of different activities (see Figure 6.4(a)). The central business district, which forms the core of the onion-ring structure, is the focus of commercial, social, and civic activities as well as the transportation system. Outside the central business district is a transition zone where residential and light manufacturing activities

Figure 6.4 CONCENTRIC ZONE, SECTOR, AND MULTI-NUCLEI STRUCTURES OF A CITY



(a) Concentric zone structure **(b)** Sector-structure **(c)** Multi-nuclei structure

are found. Further out, there is the zone for blue-collar workers' homes. Then comes the ring of higher income residences, including the better apartments and single-family dwellings. Finally, on the outer fringe is the commuters' zone in which suburbs and satellite towns are found.

The **sector city structure,** suggested by Hoyt as far back as 1939, is a modified version of the above in that it incorporates transportation factors more explicitly. The influence of transportation routes in guiding urban growth is modeled along corridors (or sectors) in addition to the ring structure (see Figure 6.4(b)). This structure recognizes that growth occurs along transportation routes since they facilitate the movement of people and freight. Also incorporated is the fact that high-income residential activities tend to move out from the city center as a result of better transportation to and from the central business district.

Finally, the **multi-nuclei structure** recognizes that a disparate group of centers grow to merge into a multi-nucleated urban area. Certain heavy industries may have located themselves in certain parts of town. At the same time mutually exclusive facilities are likely to separate one from the other. Thus quality residences tend to locate themselves away from the industries for environmental reasons. As a result, several central places evolve within the same urban area. An illustration of this concept is again shown in Figure 6.4(c).

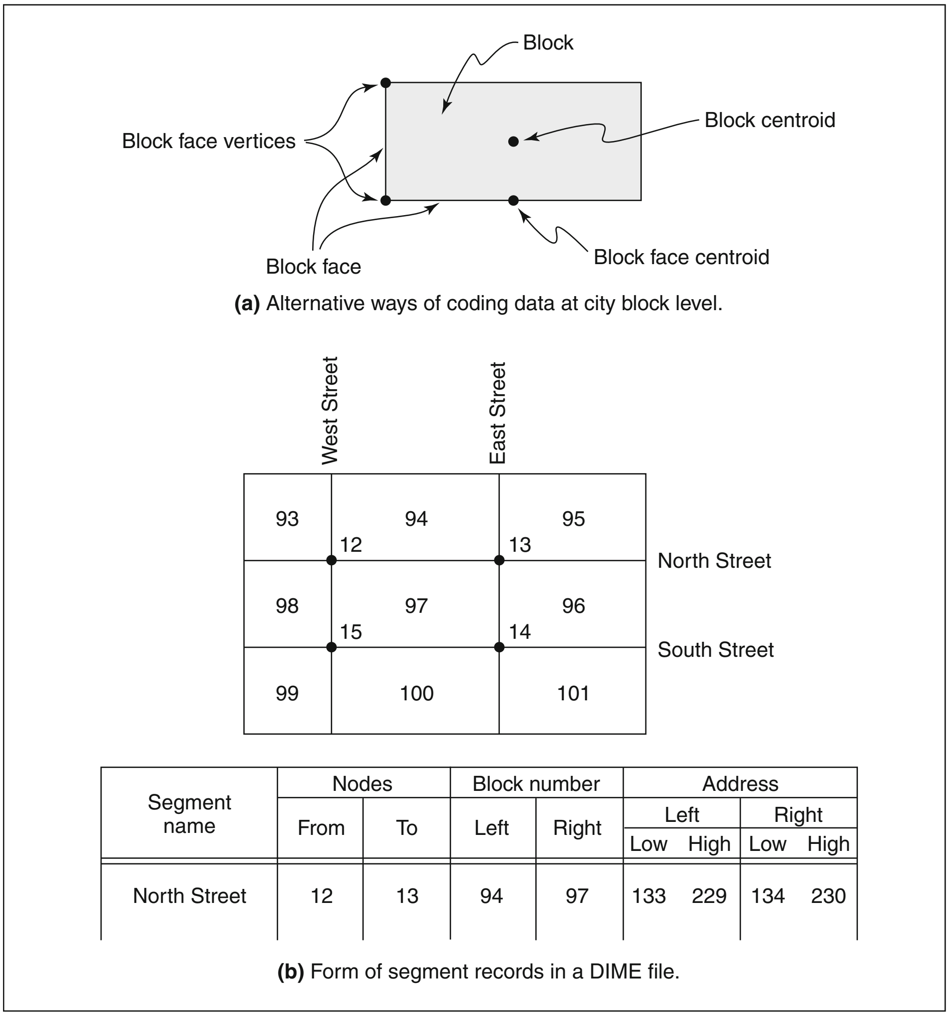
There are other postulated geographic structures of a city, but those described above represent the classic ways to classify a city into subregions for geographic reference. It would seem that a GIS should be able to take into consideration such a classification scheme. In reality, however, most GIS tend to organize their classification scheme along arbitrary geographic boundaries of census tracts, school districts, voting precincts, and traffic zones. They bear little resemblance to the logical scheme as outlined above. In recent years, districting models have been proposed to divide a community into logical subdivisions (Benabdallah and Wright 1992; Bennion and O'Neill 1994; Ahituv and Berman 1988). This represents a revived interest in the structure of human settlements. More will be said on district clustering in Section XII of this chapter.

C. Dual Independent Map Encoding System

The fundamental urban data classification used in practice goes to a level of detail way beyond the internal city structure schemes above. In a way, it also goes beyond the census tracts, school districts, voting precincts, and traffic zones. In the United States, the requirement for a continuing metropolitan planning process in the 1960s created an increased demand upon the Bureau of the Census to provide a small area data and expanded data-user services. This, in turn, led to the initiation of the Census-Use Study, which was to improve methods for relating census data to local agency data at a fine geographic scale. By the 1970s, a fairly standardized universe of urban geographic base files utilizing the Dual Independent Map Encoding (DIME) system, based on block-face coding, had been implemented in almost all standard metropolitan statistical areas.

The 1970 census instituted a computerized procedure that incorporated many of the advances developed in the studies in the 1960s. As mentioned, the major geocoding innovation was DIME. A DIME geographic base file is essentially a description of block boundaries defined by its nodes (or vertices). Figure 6.5 illustrates this technique through a series of illustrations. Many of the steps in the traditional geocoding process are eliminated by the use of specially prepared

Figure 6.5 ILLUSTRATING THE DIME FILES



(a) Alternative ways of coding data at city block level.

Segment name	Nodes		Block number		Address			
					Left		Right	
	From	To	Left	Right	Low	High	Low	High
North Street	12	13	94	97	133	229	134	230

(b) Form of segment records in a DIME file.

SOURCE: Hutchinson (1974). Reprinted with permission.

master-coding-maps and the keying of census files to coding maps through the geographic identifiers. The construction of noded map, map-resources lists, block numbering-schemes, field work and Address Coding Guide (ACG) preparation may now be unnecessary. Use of standard DIME/ACG files entails the construction of the arterial network in the DIME system, the reconstruction of census data to fit traffic zones, and the addition of local transportation and land use files to the modified DIME Census-file. The arterial network can be reconstituted from DIME. To do this, however, a DIME file must be adjusted for transportation-system applications. That task will include creating a new network file, adding data on traffic direction, capacity, pavement width, etc. (See Figure 6.5(b)). To reconstruct census data by traffic zones, it is necessary to provide a table of equivalents between census areas and traffic zones. Similarly, the addition of local codes allows use of the DIME file in the analysis of local and census data as they relate to local areas.

The use of DIME files is based on the fact that the city block is one of the smallest, most standard and relatively permanent urban areal units. Block face is the lowest common-denominator unit for urban geographic base files because, in general, cities are made up of blocks. With data gathered and recorded by blocks, data sets can be aggregated or disaggregated to conform to any number of special area boundaries, for instance, school districts, traffic zones, or police precincts. Thus, only one geographic coding system is required to meet the varied demands of many users.

D. Topologically Integrated Geographic Encoding and Referencing

Unlike the urban environment, there is no analogous common-denominator unit for national geographic base files. A nation's geography encompasses a far more heterogeneous mix of land use patterns, natural areas, governmental entities, and other spatial orderings than that which characterizes metropolitan geomorphology. While many national geocoding systems are based on county units or units compatible with county boundaries, there are important exceptions, such as zip code zones and congressional districts. These do not necessarily aggregate to the county level. There is a lack of definition, both semantic and geographic, of sub-county units. There is the problem of variation between urban and rural land use, settlement patterns, and population densities which create great disparities in the size of the spatial units coded both within a single system and among the various systems (Schweiger 1992; Gryder 1992).

As part of the 1990 census, the U.S. Department of Commerce, Bureau of the Census, developed an automated geographic database, known as the Topologically Integrated Geographic Encoding and Referencing (TIGER) system. TIGER provides coordinate-based digital-map information for the entire United States, Puerto Rico, the U.S. Virgin Islands, and the Pacific Territories over which the United States has jurisdiction. The TIGER system has significantly improved the accuracy of the 1990 census maps and geographic reference products. Extract files from the TIGER system permit users with appropriate software to perform such tasks as linking the statistical data in the 1990 Census of Population and Housing: displaying selected characteristics on maps or a video display screen at different scales and with whatever boundaries they select for any geographic areas of the country. For example, a map for a particular county may be displayed showing the distribution of the voting age population by city block. The Bureau makes the information, called TIGER/Line™ files, available to the public on CD-ROM. A program, available from some of the most widely circulated GIS software, allows the users to display the information in a graphic form.

TIGER data is the most widely used spatial data used to geographically define a local area or region available today. They replace the 1980 GBF/DIME (Geographic Base File/Dual Independent Map Encoding) files, and contain these data elements:

1. Census map features such as road, railroad and rivers
2. Feature names and classification codes
3. Alternate feature names
4. Associated 1980 and 1990 census geographic area codes
5. Federal Information Processing Standard (FIPS) codes
6. Latitude and longitude coordinates
7. For areas formerly covered by DIME files: address ranges and zip codes

Other TIGER-related products that may be helpful for specific applications include:

1. TIGER/DataBase™—containing point, line, and area information from TIGER's internal data base, including additional information not available in the TIGER/Line™ files;
2. TIGER/Boundary™—containing coordinate data for specific 1990 census tabulation area boundary sets; for instance, a file containing all state and county boundaries, and another containing all census tract and block-numbering area boundaries;
3. TIGER/Tract comparability™—providing information for 1980 and 1990 census tracts.

Klosterman (1991) gives an excellent introduction to the TIGER system, emphasizing applicational considerations. Also included are a glossary of terms and contacts for further information.

E. Other Data Sources

In the United States, data from National Aeronautics and Space Administration (NASA), National Oceanic and Atmospheric Administration (NOAA), and Department of Interior through the U.S. Geological Survey (USGS) have been particularly important supplements to data from the Bureau of Census. Specific examples of data here included remotely sensed, land use, land cover, and digital elevation data (Star and Estes 1990; Schweiger 1992). Table 6.2 presents examples of digital data sets, produced on a routine basis, that are available (or are being made available) from the U.S. government. Data from USGS can be used to define street networks. USGS offers Digital Line Graphs (DLGs) through the National Digital Cartographic Data-Base (NDCDB). DLGs are files of cartographic data primarily made by digitizing point locations and line and polygon outlines from map separation materials. (See example in Figure 6.11.) The spatial data are topologically structured. Spatial relationships, such as adjacency

Table 6.2 DIGITAL DATA AVAILABLE FROM THE UNITED STATES GOVERNMENT

Data Type	Data Source
Topography: Digital elevation model Digital terrain data	U.S. Geological Survey (National Mapping Division) Defense Mapping Agency
Land use and land cover: Ownership and political boundaries Transportation Hydrography	U.S. Geological Survey (National Mapping Division) Note: Department of Energy also has transportation data
Socioeconomic and demographic data: Census tract boundaries Demographic data Socioeconomic data	U.S. Department of Commerce (Census Bureau)
Soils	U.S. Department of Agriculture (Soil Conservation Service)
Wetlands	U.S. Fish and Wildlife Service
Remotely sensed data	National Aeronautics and Space Administration National Oceanic and Atmospheric Administration

SOURCE: Star and Estes (1990). Reprinted with permission.

and connectivity among data elements, are explicitly encoded. In addition, DLG data elements may have coded attributes. An improved data model, called Digital Line Graph-Enhanced (DLG-E), will be available soon. DLG-E provides for the explicit representation of individual cartographic features, such as roads, counties, buildings and streams, in addition to the topologically structured spatial data provided in the current DLG. This enhancement also provides a more extensive set of attributes and relationships for these features than exists in a DLG. Other data which are available from USGS include digital elevation model data, land use and land cover data, and geographic names data, as suggested earlier and shown in Table 6.2. Remote-sensing data will be discussed in a later section of this chapter.

There is a long-range effort in the U.S. Government to create a NDCDB. This is based on the work of an interagency coordinating committee, to set standards for the format and content of digital spatial data throughout the government. The layers to be included in this database include hypsography (topographical relief), hydrography (surface water for navigation), land surface cover, surface features including vegetation, boundaries, positional control, transportation, other man-made structures, and the Public Land Survey System. One commercially available source of spatial data is EtaMaps®, available through Etak, Inc. They contain centerline street data, address ranges, political and statistical boundaries, and zip codes. They come in two formats: as ASCII format which can be read by the leading GIS software products such as ARC/INFO, AutoCAD, IGDS, INFORMAP and others and a compressed format, making EtakMaps® usable with other Etak software products. Many of the GIS software vendors provide data sources as well. We will survey these GIS software in a later section.

The United Nations Environment Programme (UNEP) is a spatial data user as well as a producer. Through the newly established Global Resources Information Database, with existing centers at the UNEP offices in Geneva, Switzerland, and Nairobi, Kenya, efforts are under way to collect and disseminate important spatial data sets for the globe, as well as provide certain kinds of assistance in spatial data collection and processing to less-developed countries. Sample data sets in the archives now include range and endangered species distribution for parts of the world, as well as small scale global data sets of soils and vegetation.

Shaw, Maidment, and Arimes (1993) reported a computer-based regulatory information system for site planning. The concept of jurisdiction is used to separate the regulations and permit requirements applicable to a particular development from those that are not. It is a useful tool for providing early feedback to prospective permit applicants. In sites with rapidly changing regulations, updating and maintaining current information may require substantial effort. To use GIS in concert with regulatory information is a feasible solution, although this has its cost implications as well. Finally, this concept can be carried over to hazardous waste regulations, environmental permitting, and appropriative water-right laws.

III. GEOGRAPHIC INFORMATION SYSTEMS (GIS)

Computer-based GIS is characterized by its ability to integrate layers of spatially oriented data through a variety of analytical approaches. The end result, if carefully executed, is productive sharing of information for multiple problem solving. Among the general advantages of GIS are (Lee and Zhang 1989):

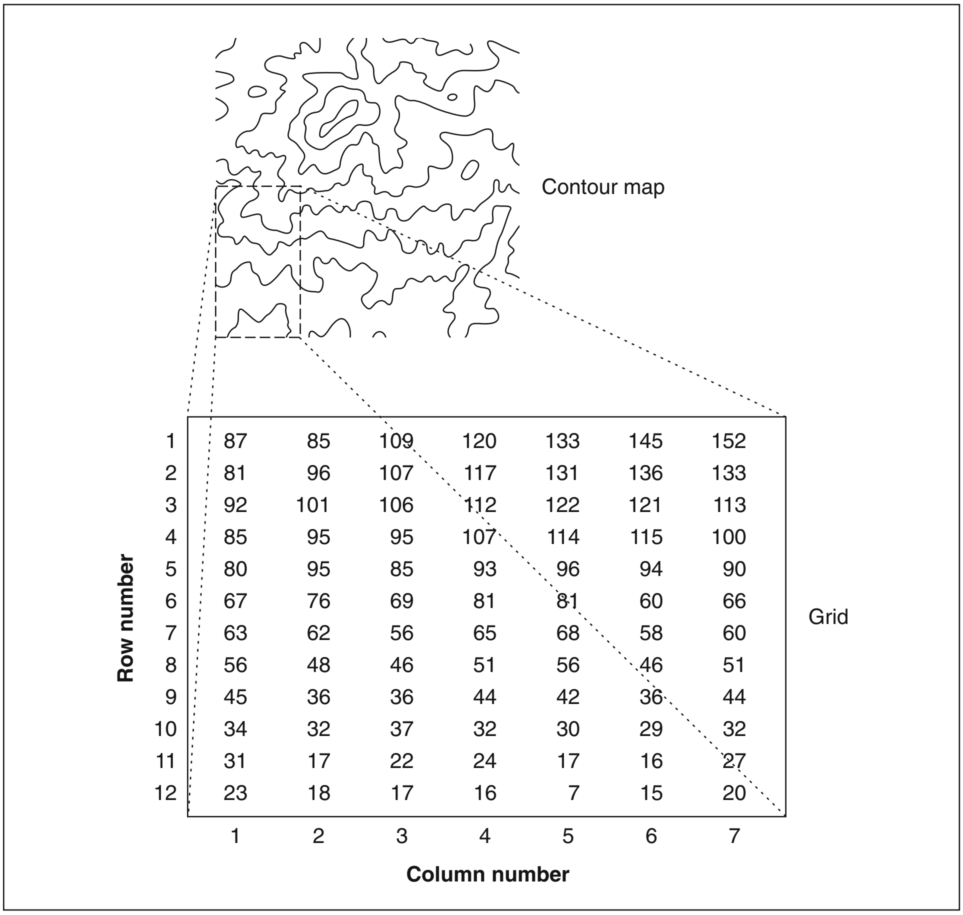
(a) The ease of data retrieval;
(b) The discovery and display of information gained by observing interaction between location and land use attributes;
(c) The capacity to process a large amount of data for spatial evaluation;
(d) The ability to make scale and projection changes, remove distortions, and perform coordinate rotation and translation; and
(e) The analysis of spatial relationships through the application of empirical and quantitative models.

A. Data Organization and Structure

The choice of a particular **spatial data structure** is one of the important decisions in designing a GIS (Star and Estes 1990). Each type of spatial data or theme in a GIS is referred to as a data layer or data plane. In each of these data layers, there are three primitive geometrical entities to encode: points, lines, and polygons or planes. **Points,** such as the locations of oil and water wells, and **lines,** such as the centerlines of roadways or streams, are key elements of this breakdown. When we consider bounded regions, such as the borders of a subdivision or the edges of a lake, we often focus on the boundary lines called the enclosed region **polygons.** We use the term to include curved boundaries in addition to straight line boundaries. Not all GISs can work directly with curves as such. More often than not, they permit a curved line to have interior digitized points in addition to the end points. Thus a curve is approximated by straight line segments. Besides geometric information, equally important is the nonspatial or attribute data. For a simple spatial object such as a well, the essential spatial information is the geodetic or geographic location of the well. Ancillary information may include its depth, date of drilling, production volume, ownership, and so forth. (See examples in Figure 6.11 and Figure 6.12.)

1. Raster Data Structure. The data structure of a GIS can be broadly classified into two types: raster and vector. In a raster structure, a value for the parameter of interest, for example, elevation above datum, land use class, and plant biomass density, is developed for every cell in a grid over space. In Figure 6.6, elevation in meters has

Figure 6.6 RASTER DATA STRUCTURE EXAMPLE

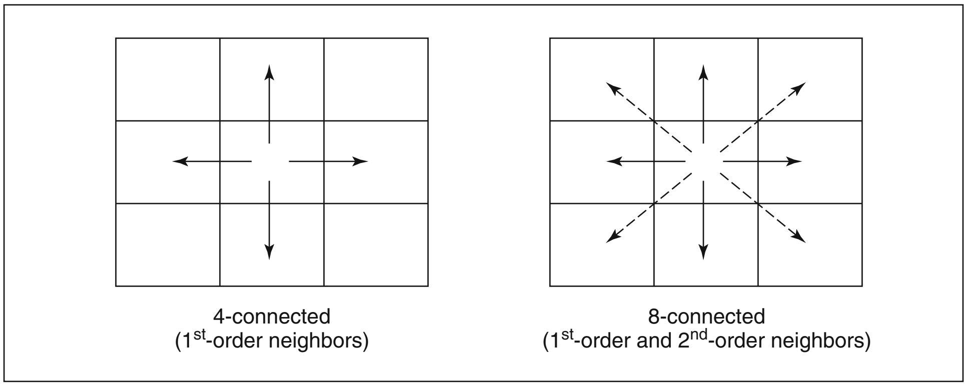


Row number	1	2	3	4	5	6	7
1	87	85	109	120	133	145	152
2	81	96	107	117	131	136	133
3	92	101	106	112	122	121	113
4	85	95	95	107	114	115	100
5	80	95	85	93	96	94	90
6	67	76	69	81	81	60	66
7	63	62	56	65	68	58	60
8	56	48	46	51	56	46	51
9	45	36	36	44	42	36	44
10	34	32	37	32	30	29	32
11	31	17	22	24	17	16	27
12	23	18	17	16	7	15	20

Column number

SOURCE: Star and Estes (1990). Reprinted with permission.

Figure 6.7 DEFINITION OF SPATIAL NEIGHBORHOOD



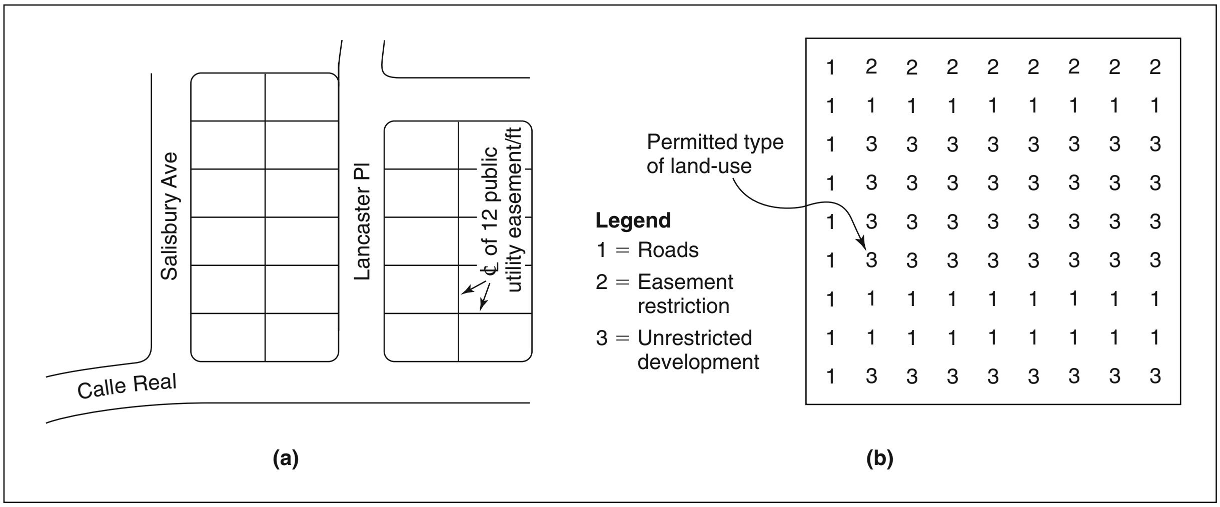
SOURCE: Star and Estes (1990). Reprinted with permission.

been recorded from a contour map on a regular grid, where each cell is referenced by the row and column numbers. Thus at the position represented by the first cell (1, 1), the land is 87 meters (290 feet) above sea level, and so forth.

One consequence of this grid system is that a cell has either four or eight adjoining neighbors, depending on one's preference, as shown in Figure 6.7. Notice that the 4-connected neighbors are closer than the 8-connected neighbors inasmuch as the diagonal elements are 1.41 times further away than the immediate 4-connected ones. The former is called **first-order neighbors** and the latter **second-order neighbors.** (See the "Spatial Time Series" chapter in Chan [2005] to see how the order of neighbors affects spatial analysis.)

Consider a development as shown in Figure 6.8(a). A map from a local planning agency shows the legal property boundaries, streets, and restrictions on construction and development due to easements of public utilities. Figure 6.8(b) shows a raster converted representation of the map. The numbers in each cell

Figure 6.8 A SUBDIVISION MAP AND ITS RASTER REPRESENTATION


Salisbury Ave
Lancaster Pl
℄ of 12 public utility easement/ft
Calle Real
(a)
Permitted type of land-use
Legend
1 = Roads
2 = Easement restriction
3 = Unrestricted development
(b)

1	2	2	2	2	2	2	2	2
1	1	1	1	1	1	1	1	1
1	3	3	3	3	3	3	3	3
1	3	3	3	3	3	3	3	3
1	3	3	3	3	3	3	3	3
1	3	3	3	3	3	3	3	3
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	3	3	3	3	3	3	3	3

SOURCE: Star and Estes (1990). Reprinted with permission.

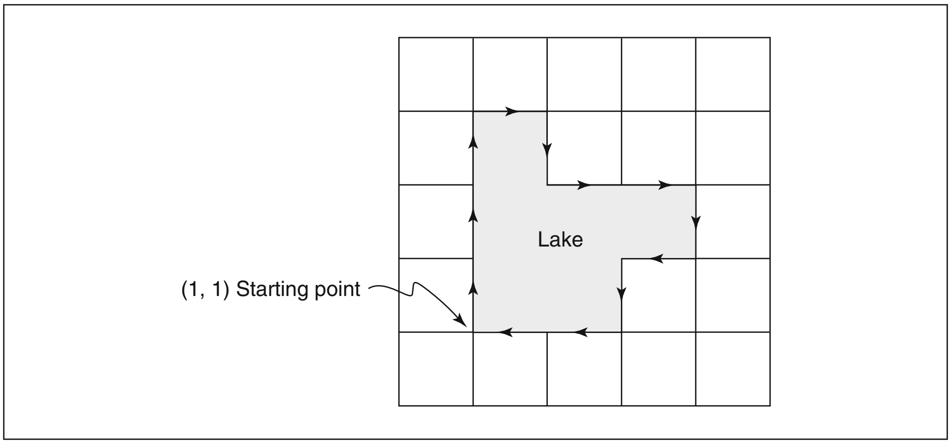
indicate the permitted land use for each cell, using a majority rule in case of ambiguity due to the coarseness of the grid overlay on the subdivision map. By adding up the number of cells in each category, we can determine the percentage area coverage of each land use category:

Land use category	*Class*	*Total cells*	*Percent of total*
Roads	1	33	41
Easement-restriction	2	8	10
Unrestricted-development	3	40	49

Raster data sets in practice can be very large. When dealing with such large data sets, there are several algorithms used to compress the data. Aside from the obvious method of increasing the coarseness of the grid, one way of compressing the data uses chain codes. **Chain codes** consider a map as a set of spatially referenced objects placed on top of a background. The coordinates of a starting point on the border of an object (for example, a lake) are recorded, and then the sequence of cardinal directions of the cells that make up the boundary are stored. As shown in Figure 6.9, the shaded area is represented, beginning from the starting point (1, 1), by 3 units north, 1 east, 1 south, 2 east, 1 south, 1 west, 1 south, and 2 west. This may be an efficient way to store areas, particularly since each spatial object is kept as a separate entity in the data base. However, some kinds of processing will be required so that the entire raster array can be reconstituted, a complex task that may amount to an unacceptable cost.

2. Vector Data Structures. The second major type of data structure in a GIS is the vector format. In a description of spatial data based on vectors, we make the assumption that an element may be located at any location, without the positional constraints of a raster array. Vector data structures are based on elemental points whose locations are known to arbitrary precision, in contrast to the approximate raster data structures described above. As a simple example, to store a circle in

Figure 6.9 A CHAIN CODE REPRESENTATION



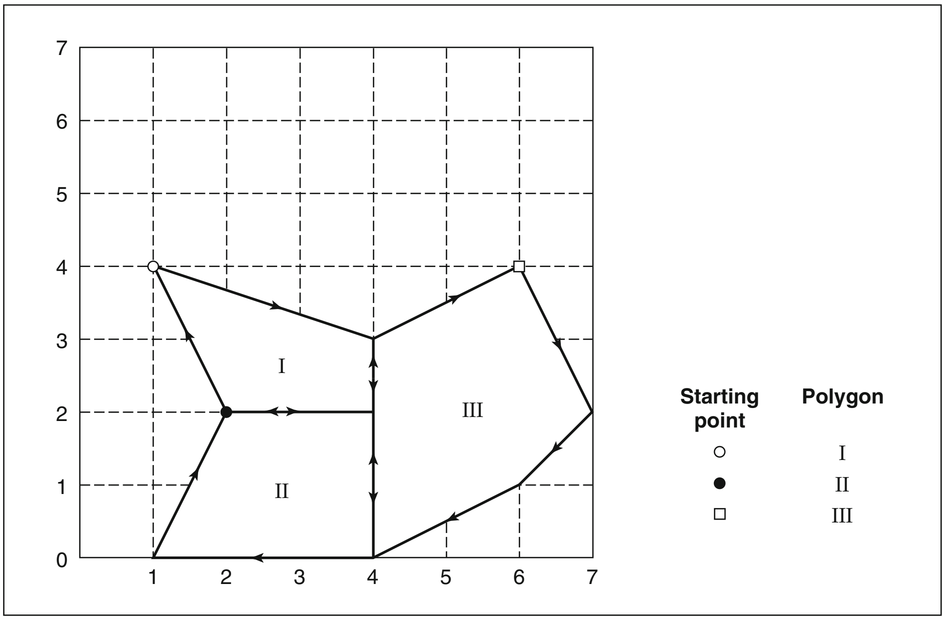
SOURCE: Star and Estes (1990). Reprinted with permission.

one of the raster data structures, we might find and encode all the raster cells whose locations correspond to the boundary of the circle. This is often called a low-level description of the circle. A high-level description, on the other hand, might efficiently store the circle by recording a point location for the center of the circle, and specifying the radius. In this example, the high-level description based on a vector representation is more efficient in terms of the amount of data required, as well as more precise.

Several forms of vector data structures are in common use. In a **whole polygon structure,** each layer in the data base is divided into a set of polygons such as the one shown in Figure 6.10. Each polygon is encoded in the data base as a sequence of locations that define the boundaries of each closed area in a specified coordinate system (sometimes called a boundary loop). Each polygon is then stored as an independent feature. There is no explicit means in this system to reference areas that are adjacent. This is, to some extent, comparable to the chain-coded raster discussed above, in that for both a whole polygon structure and a chain-coded raster, the emphasis is on the individual polygonal areas, where each discrete area is stored separately. Thus the three regions in Figure 6.10 appear as

Polygon I	*Polygon II*	*Polygon III*
1, 4	2, 2	6, 4
4, 3	4, 2	7, 2
4, 2	4, 0	6, 1
2, 2	1, 0	4, 0
		4, 2
		4, 3

Figure 6.10 EXAMPLE OF A WHOLE POLYGON STRUCTURE



SOURCE: Star and Estes (1990). Reprinted with permission.

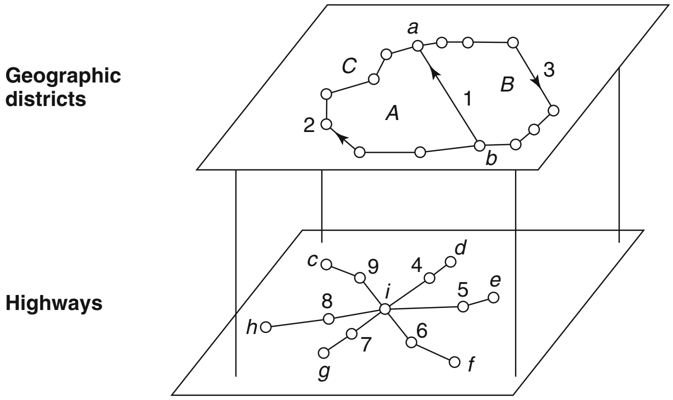
Attributes of the polygons, such as land cover or ownership, may be stored with the coordinate list. Please note that by maintaining each polygon as a separate entity this way, the topological organization of the polygons is not maintained. By topology is meant the relationships between different spatial objects: which polygons share a common boundary, which points fall along the edge of a particular polygon, and so on. In a whole polygon structure, line segments that define the common edges of polygons are recorded twice, once for the polygon on each side of the line. Similarly, points that are shared by several polygons, such as location (4, 2) in the example, will also be represented several times in the data base. With this organization, editing and updating the database without corrupting the data structure can be difficult.

One of the best known standard vector file formats is the previously mentioned DLG of the USGS. The agency is producing these vector files based on the source materials used to compile the USGS 7.5-Minute and 15-Minute Topographic Maps Series. A separate set of DLG data files is based on 1:2,000,000-scale map products. The data contents of the DLG files are subdivided into different thematic layers. One layer consists of boundary information, including both political and administrative boundaries in the region. A second layer is for hydrographic features. A third layer is for the transportation network in the area. Finally, the fourth layer is based on the Public Land Survey System, which has as its focus a survey system administered by the U.S. Bureau of Land Management.

The essential data elements of the DLG level-3 structure are similar to the other vector data structures discussed in connection with DIME files. Nodes represent either end points of lines or line intersections, while additional points are used where required to indicate significant features along lines. Lines have starting and ending nodes, and as such, they permit us to specify a direction along the line as well as the areas on the left and right sides of the line. A special degenerate line is defined as a line of zero length and is used to define features that are indicated on the map as a point. Degenerate lines are recognizable because they have the same starting and ending node. Areas in the DLG format are completely bounded by line segments. Each area may have an associated point that represents the characteristics of the area; the point location is arbitrary and may not even be within the area. The point, line, and area elements provide information about topology and location. In addition, there is an extensive system for coding attribute information for the elements. The attribute codes are based on those features represented on USGS topographic maps. Attribute codes are structured in a specified way, with both major and minor code components, where a major code may signify surface cover for example and a minor code may contain more specific descriptors.

3. Relational Vector Structure Graphical approaches utilize cartographic principles—for example, symbols, line weights, and color—for characterizing spatial features and their type and magnitude. Relational databases contain an ordered set of information grouped together in two-dimensional tables known as relations. Users define the relation that is appropriate to the query whether the tables are already available or need to be constructed by the controlling program. Relational databases have the advantage that their structure is very flexible and can meet the demands of all typical queries that can be formulated. Figures 6.11 and 6.12 present examples of such a data structure. Illustrated in these examples are representations of areas, lines, and points, as well as their attributes, stored as alphameric mirror

Figure 6.11 CHAIN AND POLYGON DATA RECORDS FOR GIS



Geographic districts – Polygons

Area ID	Ring (chain list)*
A	−1, 2
B	3, 1

Geographic district – boundary chains

Boundary chain ID	From node	To node	Left area ID	Right area ID	*x-y* coordinate
1	*b*	*a*	*A*	*B*	$x_1y_1 \ldots x_ny_n$
2	*b*	*a*	*C*	*A*	$x_1y_1 \ldots x_ny_n$
3	*a*	*b*	*C*	*B*	$x_1y_1 \ldots x_ny_n$

Highway network chains

Network chain ID	From node	To node	*x-y* coordinates
4	*i*	*d*	$x_1y_1 \ldots x_ny_n$
5	*i*	*e*	$x_1y_1 \ldots x_ny_n$
6	*i*	*f*	$x_1y_1 \ldots x_ny_n$
7	*i*	*g*	$x_1y_1 \ldots x_ny_n$
8	*i*	*h*	$x_1y_1 \ldots x_ny_n$
9	*i*	*c*	$x_1y_1 \ldots x_ny_n$

*Negative entries indicate reverse order

SOURCE: Nyerges and Dueker (1988). Reprinted with permission.

images of the graphical counterpart. The example in Figure 6.12 shows the related highway network attributes, including travel times and traffic volumes. In summary, these are the advantages of a relational organization:

(a) All data structures can be normalized (such as a unit-square representation of a rectangular map)[2];

(b) Spatial consistency is insured across entities (as key points are geocoded in *x-y* coordinates);

Figure 6.12 LOCATIONAL DATA LINKED TO ATTRIBUTE DATA

Highway-network chains

Network chain ID	From node	To node	Coordinates
4	i	d	$x_1y_1\ldots x_ny_n$
5	i	e	$x_1y_1\ldots x_ny_n$
6	i	f	$x_1y_1\ldots x_ny_n$
7	i	g	$x_1y_1\ldots x_ny_n$
8	i	h	$x_1y_1\ldots x_ny_n$
9	i	e	$x_1y_1\ldots x_ny_n$

Locational data

Highway-network data

Chain ID (control section ID)	Travel time (Min.)	Traffic volume (veh./day)	Other attributes
4	15	5,000	…
5	20	10,000	…
6	17	4,000	…
7	14	6,000	…
8	22	11,000	…
9	18	3,000	…

Attribute data

SOURCE: Nyerges and Dueker (1988). Reprinted with permission.

(c) Spatial relations are compact (see, for example, the district specification in Figure 6.11) and isolated from non-spatial relationships (in Figure 6.12);

(d) Features can be attached to multiple spatial entities inasmuch as these entities are referenced against one another; and

(e) Multiple geobases can be attached to the same application database.

B. Location Reference System and Data Structure

As seen from Figures 6.11 and 6.12, GIS queries can be spatial, non-spatial (attribute), or a combination of the two through the establishment of suitable linkages. When spatial information is desired, data must be established within a coordinate system that can serve as a spatial reference system. In addition to the reference control scheme, data are generally related to a map base that maintains good horizontal and vertical control. USGS 1:24,000- and 1:62,500-scale quadrangles have been employed in a number of GIS studies. The USGS 1:100,000-scale maps are gaining popularity and utility because of the useful scale for a variety of small-scale GIS investigations and because of the current U.S. Census TIGER files that utilize data from that map series. USGS map series provide, in general, good topographic, transportation, hydrographic, political boundary, and spatial control on each base map to serve as the locational reference for GIS analyses. Global Positioning Systems, afforded through satellite technology, are gradually providing a mechanism to secure accurate survey coordinate information with time and cost savings.

As mentioned, closely related to location referencing is topological information. Topological information allows one to describe not only an object's position, but also its spatial relationships with respect to neighboring objects. Some kind of topological information is implicit in spatial data. In a simple raster structured data file, for example, there is a specified spatial organization for the data. The regularity in the array provides an implicit addressing system. This permits rapid random access to specified locations in the database. Thus we know immediately those cells that are adjacent to any target location, and we can easily find and examine those regions that bound a specified group of cells. Topological information in vector structures is often coded explicitly in the database. Line segments with DIME files, for example, have identifiers and codes for the polygon on either side. When topological relationships are not explicitly coded in vector data structures, it can be relatively expensive and time-consuming to constitute them.

The advantages and disadvantages of raster versus vector spatial data structures hinge on data volume (or storage efficiency), retrieval efficiency, robustness to perturbation, data manipulation efficiency, data accuracy, and data display. While some of these have been discussed in the previous section, there are fundamental differences between the two systems that make comparison irrelevant: raster is quasi-continuous, while vector is clearly discrete; raster representation may be considered more dense than vector because more unique values are stored. On this basis, the two systems are geared toward different applications, and they have their respective roles to play. To illustrate how difficult the comparison job really is, consider comparison of processing efficiency in modern GIS systems. Traditionally, overlay operations are thought to be more efficient in raster systems. In current data processing technology, however, there may be an efficient means to determine the approximate locations of polygons by maintaining a separate index database. Using such an index to structure a search through the spatial data, a comparison of raster and vector data structures based on processing speed may be more sensitive to the spatial data itself than to the choice between the two data structures. If forms of both raster and vector structures are found in a GIS, as well as structure conversion routines and appropriate analysis tools for each data type, then the data could be stored in their natural form to both optimize geographic specificity and minimize conversion costs and attendant bias. This also permits analytic procedures to operate on a data structure where efficiency or accuracy is highest. While this strategy is more complex than one in which all data are stored and manipulated in a single data structure, efficient software and hardware for vector/raster conversion can significantly reduce the size of the problem.

Because of today's prevailing philosophy of data sharing between various organizations, federal agencies have begun distributing spatial data using Topological Vector Profile (TVP) as part of the Spatial Data Transfer Standard (SDTS). The most notable of these applications is USGS's conversion of all 1:100 000-scale and 1:2 000 000-scale DLG-3 data to TVP and making TVP available free of charge on the Internet. Lazar (1996) provided a primer on using the TVP. SDTS will eventually cover all aspects of spatial-data transfer, including the conceptual modeling of spatial data itself. These encompass the definition of 32 vector and raster spatial objects. SDTS would have specifications for data quality reports, logical specifications for transferring data (what items can be transferred and how they are organized), and the physical field format of the data transfer. Currently, TVP requires several spatial objects to exist in every data set. One feature of TVP is that it provides a common dictionary to unify hitherto diverse spatial object terminologies. For example, the following spatial objects are defined:

(a) **planar node:** a zero-dimensional object that is a topological intersection or endpoint of one-dimensional objects,

(b) **complete chain:** a one-dimensional object that references starting and ending nodes and left and right two-dimensional objects,

(c) **GT-polygon:** a two-dimensional object, where the GT stands for geometry and topology, and

(d) **universe polygon:** the special GT-polygon that covers the rest of the universe outside of other GT-polygons; there is always exactly one universe polygon.

An SDTS data set is referred to as a transfer. A **transfer** consists of a group of files encoded. In the TVP, all files for a particular transfer will be in a single, separate directory for any medium with a directory structure. There is also an ASCII text README file associated with it. Part of the file name refers to a logical grouping of related information. For example, it may facilitate the transfer of one-dimensional spatial objects such as complete chains. It should be noted the system is still evolving and additional features are being implemented, often at the suggestion of the users.

C. Geospatial Metadata

To further facilitate transferability, metadata is adopted in GIS. Metadata is defined as structured information that enables a dataset to be identified, used, manipulated, and cataloged (Galati 2006). Metadata is often referred to as "data about data." The datasets to be identified include not only GIS, but also images, documents, maps, library records, and anything else searchable. Metadata is sometimes embedded within the dataset, or it could be a separate document. Either way, metadata makes the dataset "visible" to searches.

Metadata helps GIS users understand the numerous parameters surrounding the datasets. Users can quickly discern the dataset's level of precision and usefulness in tandem with other datasets and objects. Well-detailed geospatial metadata directs users on how best to use the data and to what lengths the dataset creator went to construct the resource. Through metadata review, an astute GIS user could identify which dataset resources are of high quality, which are beneficial for his/her application, and which are irrelevant.

Geospatial metadata has proven to be particularly valuable in the technical and scientific community. New, refined definitions of geospatial metadata are concurrently being constructed to better define and catalog the available technical data bases. Many worldwide organizations feel that geospatial data should be standardized for swift interoperability and exchange. Standards provide universal terms for the various datasets they describe. Universal terminology presents the capability of automatic searches for specific terms. Thus standards enable simpler information interchange.

Certain standards are mandated by governments and are required by local agencies. For example, the U.S. government requires all agencies to use the Federal Geographic Data Committee (FGDC) content-standard, while the Australian Government requires the Australian government Locator Service standard. Irrespective, geospatial metadata standards offer the GIS user an efficient way to organize and catalog available datasets.

In the 1990s, the U.S. government became very interested in the retention, organization, and dissemination of geospatial data. By this time, GIS, Global Positioning System (GPS), and satellite imagery were being used in full force. The FGDC took the first steps toward a comprehensive National Spatial data Infrastructure (NSDI). In 1994, NSDI was officially formed and a mandate for the FGDC to create geospatial data standards was enacted. FGDC's Content-Standard for Digital Geospatial Metadata was essentially born.

In 2003, another mandate would have the NSDI provide an infrastructure through which data producers and users could share geospatial data. Partnerships were formed with public and private data producers to increase data availability to geospatial data users. A metadata clearinghouse was formed, serving as an online or offline reservoir of published data that is being offered to the public. Oftentimes, these geospatial metadata clearinghouses also offer geospatial datasets.

IV. REMOTE SENSING SYSTEMS

GISs are demonstrably powerful tools for the management and analysis of spatial data. Remote sensing systems are equally powerful tools for the collection and classification of spatial data. However, nearly all of the currently operational GISs utilize maps as their primary source of spatial data. These complex documents, designed for visual search and retrieval by human operators, are digitized (usually manually) and then entered into the master spatial database of the GIS. Although many of the maps used as input are derived from aerial photography or occasionally other remote sensing devices, there is little use of digital remote platforms as a direct data input. The last few years have seen an increased interest in the direct use of remote sensing data as inputs to GIS, but much of this interest has been centered in the remote sensing community rather than among the potential primary users, those who make operational use of GIS.

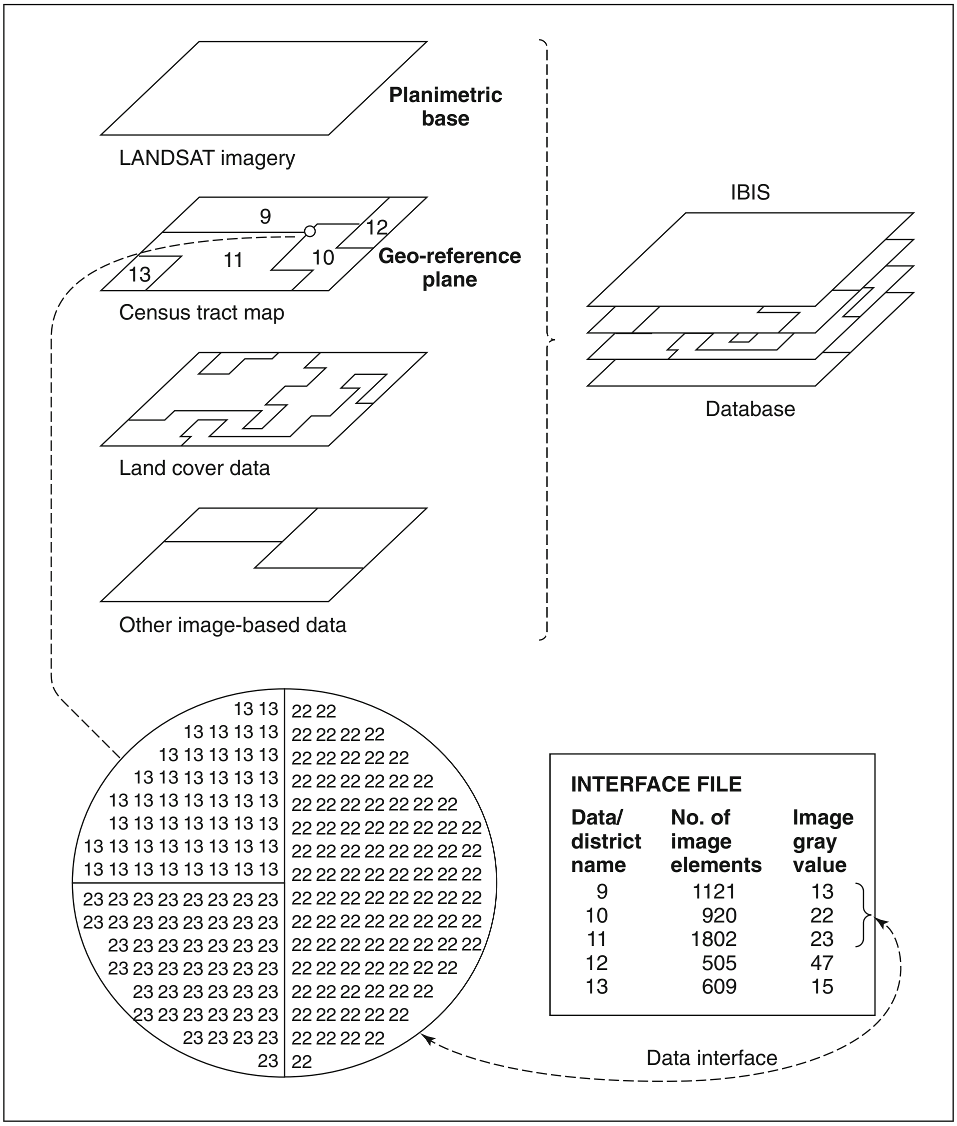
A. Interface between Remote Sensing Data and GIS

Perhaps the best way to explain how remote sensing data can serve as input to GIS and vice versa is through a case study. Computer image processing at Caltech's Jet Propulsion Laboratory (JPL) resulted in the development of an Image Based Information System (IBIS) (Marble and Peuquet 1988). Most data entered into IBIS are in raster (image-based) format. However, the system is configured in such a manner that other data types, such as graphical and tabular, may be used in analysis as well. Data input is a three-stage process. The first stage, called **data capture,** includes all operations up to the point where a data file is computer readable. Data capture costs are enormous for many basic kinds of data, such as the demographic and economic data gathered by the U.S. Bureau of the Census. Another common method of data capture is to develop a coordinate digitization of boundaries or linear features from a map. The map is not computer compatible but the digitizer output is and can be used in subsequent processing steps. In order to maintain geometric consistency between all data planes included in the database, an image plane exhibiting good radiometric and planimetric qualities is designed to be the data plane. All other data planes are geometrically corrected to register to the planimetric base. One can integrate various data types to form an IBIS database (see Figure 6.13). Since the primary data structure is a raster format, image data planes are directly entered into the system. Graphical forms of data, usually obtained in Cartesian reference form, must be transformed into image space, but are linked to the image database through a local interface, as shown in Figure 6.13. Graphical or vector data may also be entered into the IBIS database. Graphical data are either produced locally on a coordinate digitizer or are obtained from a data tape. Regardless of the data origin, graphical data are transformed into image space prior to inclusion in the IBIS database.

All tabular files (interface files) are linked to at least one of the geo-reference planes included in the IBIS database. The specific link is obtained by storing the numerical value (gray tone) representing each region of the geo-reference plane with tabular data describing attributes of that region (Figure 6.13). Attribute data may be statistical in origin, an identification code, or may be the result of an image plane comparison routine such as polygon overlay or cross-tabulation. As distinguished from the GIS discussion, remote sensing information is coded in digitized, or **pixel** (picture element) format. This avoids the referencing scheme of Figures 6.11 and 6.12 in storing lines and districts, but it usually increases the data storage and processing costs since more information is being processed.

In previous discussions, we make a distinction between raster versus vector data structures, or cellular versus organizational referencing systems. The traditional advantages and disadvantages of raster versus vector spatial data structures hinge around storage efficiency, retrieval efficiency, robustness to perturbation, data processing efficiency, data accuracy, and data display as mentioned previously. In spite of its raster format, relational data structure such as the one outlined in IBIS has the potential for efficient search among raster and vector data structures, at the expense of data file management complexity. As we have seen, such a system design permits search through either the geometrical entities or the attribute data, without the other getting in the way, since these two kinds of information are stored separately. Thus, one expects better data-retrieval performance for simple kinds of search, which should result in more efficient operations. In any event, such a system can minimize the computer's input/output

Figure 6.13 FORMATION OF AN IBIS DATABASE



INTERFACE FILE

Data/ district name	No. of image elements	Image gray value
9	1121	13
10	920	22
11	1802	23
12	505	47
13	609	15

SOURCE: Marble and Peuquet (1988). Reprinted with permission.

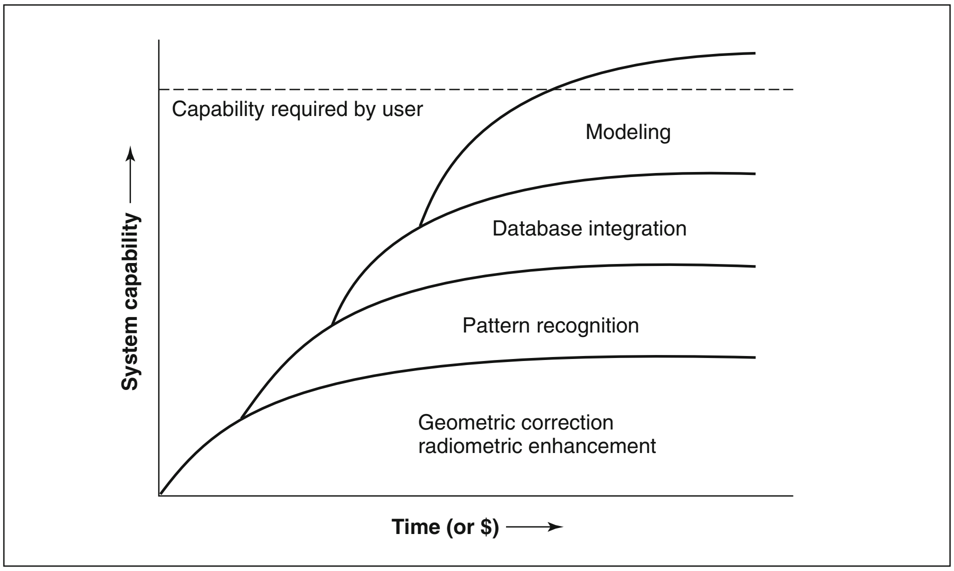
operations that are required to use the output of one search operation as the input for another. This may be particularly important when working on multi-user systems and is typically done at the expense of more complex file management.

B. An Assessment

An image-based information system is important for the full utilization of satellite imagery data. The future availability of frequent updates of land resource inventory statistics, with a known and acceptable sampling accuracy, should permit the incorporation of these data with the annual updates published by other governmental bureaus. The increasing sophistication of the use of GIS and remote sensing information can be represented by Figure 6.14. At the outset, concerns ranged over geometric fidelity and classification accuracy of satellite imagery. Since those early days, significant progress has been made. We are able to remove noise from images efficiently and delineate boundaries of lakes, forests, constructed facilities, and other land features with confidence. The capacity to perform radiometric enhancement and pattern recognition will be amply illustrated in Section VI of this chapter. The field is now embarking upon database integration and modeling activities that utilize the extended capabilities of remote sensing.

The projected demands to be placed upon GISs will put a strong emphasis on the capability to store and retrieve large amounts of data and to manipulate data sets for portions of the files efficiently. A major drawback facing most geocoding procedures is that they rely on sequential computations applied to tabular data strings and, as such, require a large investment in formatting or processing data that are inherently two-dimensional. Raster scan data-bases avoid many of these problems and possess additional advantages. The video communications field has been addressing and continues to address both the problems of mass storage and application of rapid interactive processing that place a minimal reliance upon computer software routines. The specialized requirements of GIS should derive considerable benefit from the image processing field in the future. The outlook is bright in the continuing emphasis on direct image communication. At present, the interface between GIS and remote sensing systems is weaker than it should be, and each side suffers from a lack of critical support of a type that could be provided by the other. The GIS has a continuing need for timely, accurate update of the various spatial data elements held in its system and remote sensing systems could, in many cases, benefit from access to highly precise, ancillary ground data that could significantly improve classification accuracies. In addition, there are a number of significant technical problems that would benefit from a joint, well-planned attack rather than the present disaggregated

Figure 6.14 IMAGE PROCESSING DEVELOPMENT



SOURCE: Marble and Peuquet (1988). Reprinted with permission.

and disorganized approach. A prime example is data management, since no operational database management system exists that will handle, in a cost-effective and efficient manner, the large volume of spatial data involved in both systems.

C. Remote Sensing Technology

"Photos from space" (Zimmerman 1988), a popularized term for remote sensing from satellites, represents the latest technology for collecting spatial-temporal data. Perhaps the most familiar remote sensing device is a weather satellite, which gives a resolution of one kilometer (0.59 miles) or less. The instruments normally cover areas the size of a continent in a single shot. The National Weather Service (NWS) has defined requirements for the next generation of geostationary operational environmental satellites (GOES), which NOAA has labeled GOES-Next. The emphasis in the NWS during the 1990s has been on improving short-term (0–12 hour) forecasts of severe weather events such as tornadoes, severe thunderstorms, hail, and flash floods. GOES-Next is expected to provide strong support to improving forecasts of these phenomena and will offer improvements in both imagery and vertical-temperature/moisture-sounding capabilities. Imaging capabilities of GOES-Next will be practical to an accuracy of 4 km (2.5 mi) or less.

Another major satellite is the earth-resource monitoring LANDSAT, which has a finer resolution (80 × 80 meters or 87.4 yards × 87.4 yards) on its multispectral scanner (MSS). Similar to (one km) × (one km), the 80 × 80 resolution detail is usually referred to as a pixel. Tremendous progress has been made since LANDSAT's early stages. Pixels of LANDSAT "photos" can be as detailed as 30 × 30 meters (32.8 yards × 32.8 yards) today on its thematic mapper (TM). Satellites with pixels smaller than one meter (1.09 yards) on the side are usually used for military reconnaissance. But the line between military and civilian satellites blurred when the French remote sensing satellite SPOT was able to offer commercially 10-meter (10.9 yard) resolutions in black and white and 20 meters (21.8 yards) in color. In July 1987, the then Soviets offered a 6-meter (6.54 yard) resolution satellite and are in the process of marketing 2-meter (2.22-yard) resolution (Foley 1994). Sweden is now considering a satellite offering one-meter resolution, with the purpose of arms verification in mind.

Some of these satellites can measure infrared radiation, inasmuch as most satellites have several sensors responding to a variety of spectral ranges. LANDSAT and certain Russian satellites can detect the long wavelength radiation produced by heat sources. Both LANDSAT and SPOT can detect short wavelength infrared radiation, which is produced by very hot sources such as the sun. (This includes reflection of the sun's rays by shiny objects.) SPOTs 5, 6, and 7 are planned for the decade from late 1990s through early 2000s. The latest Russian satellite with 6-meter resolution is also capable of detecting these short infrared wavelengths. The Canadian RADARSAT satellite does not have the detailed resolution of the optical image spacecraft. However, it will be able to take pictures at night and through clouds. The Japanese planned to launch a series of Advanced Earth Observing Satellites starting in the late 1990s, continuing earlier attempts at marine sensing and radar satellites. Meanwhile, the U.S. launched LANDSAT 7 successfully on April 15, 1999.

Another factor in remote sensing is the frequency of surveillance. Satellites cannot orbit the Earth faster than once every 90 minutes, since drag would otherwise draw them inside the atmosphere. A camera can photograph only a limited swath of the Earth during each revolution. Hence the best satellite would require a full day to photograph the entire Earth, considering the number of revolutions required to piece the swaths together. This means an average lag time of a half day is required to acquire a specific picture, unless geosynchronized satellites are used, concomitant with their high cost.

The most sophisticated technology has been developed for military applications. Given a two-dimensional data set, such as a satellite picture, if we have the necessary elevation information, which can be derived from a series of satellite pictures at different viewing angles, a geometric model can be constructed in the computer. The output image will be a three-dimensional representation of what started as a two-dimensional scene. This type of image manipulation has many possible intelligence and defense uses. American bomber pilots could rehearse in simulators for low-level bombing missions, becoming familiar with enemy terrain without ever going near it. In February 2000, NASA launched the Endeavour space shuttle, whose crew intended to scan 80 percent of the earth's surface. The all-weather radar image produced a three-dimensional map more accurate and comprehensive than ever before.

So far an average remote sensing satellite has typically cost $300 million (Zimmerman 1988). But with today's off-the-shelf equipment, a five-meter resolution satellite could be built and launched for less than $10 million. Total sales from satellite photography range from hundreds of millions of dollars to $7.4 billion or more, according to KRS Remote Sensing, a Kodak Company. The large range is a reflection of the uncertainty associated with the U.S. Government's national security regulation of commercial use of satellites. This regulation happens in an increasingly international and competitive market, where remote sensing service can be made readily available outside the U.S. at a reasonable cost.

In March 1994, the U.S. Administration removed restrictions on the quality of satellite photos, approving the sale of images able to reveal objects one-meter (1.11 yard) in resolution or possibly smaller (Foley 1994). Liberalizing the policy even further, manufacturers are allowed to sell foreign buyers spacecraft that are roughly equivalent to older U.S. spy satellites. However, companies selling imagery will be subject to conditions that apply to operating licenses granted by the government, conditions designed to maintain government control over the dissemination of such technology. While there are clamors about lost opportunities, license applications mount as major U.S. companies seek a share of the vast potential market. At least three U.S. organizations have 1-meter systems under development with launch dates from late 1997 to 2000 (Amato 1999; Corbey 1996). In October 1999, Space Imaging released the first commercial 1-m black and white image. Developers of the proposed systems expect the availability of high-resolution imagery to touch off a rapid increase in the size of the satellite data user market. They predict that higher resolution is exactly what is needed to convert GIS users who have not yet tried satellite or aerial imagery.

QuickBird is a high-resolution commercial earth observation satellite. It was launched in 2001 as the first satellite in a constellation of three. The companion spacecraft, WorldView-1 and WorldView-2, completed the constellation in 2009. QuickBird collects the second highest resolution commercial imagery of Earth after WorldView-1, and boasts the largest image size and the greatest on-board storage capacity of any satellite. The satellite collects panchromatic imagery at 60–70 centimeter (2 feet) resolution and multispectral imagery at 2.4- and 2.8-meter (8.53 feet) resolutions. At this resolution, details such as buildings are easily visible. The imagery can be imported as a backdrop for mapping applications, including Google Earth and Google Maps. Its acquired images can cover more than three times the area of North America in the course of a year, while its spacecraft weighs less than half as much as Landsat 7. The latest addition to the constellation, WorldView-2, is the high resolution 8-band multispectral commercial satellite. The satellite collects images at nadir with 0.45 meter (17.72 inches) resolution panchromatic and 1.84 (6.037 feet) multispectral resolution. QuickBird, WorldView-1 and WorldView-2 form a constellation offering very high revisit and large area collection capacity.

GeoEye set geospatial industry standards with the launch of IKONOS®, the world's first sub-meter commercial satellite. With the successful launch of GeoEye-1 satellite sensor in September 2008, successful completion of testing and calibration GeoEye released the satellite for commercial orders in February 2009. GIS and Computer Aided Design professionals are now able to work with satellite imagery at 0.5-meter (1.64 feet) resolution, two-meter digital raster Digital Elevation Models, one-meter (3.28 feet) elevation contours and Triangulated Irregular Networks models. This facilitates a three-dimensional computer work environment, supporting the planning and construction of roads, facilities, pipelines and many other project applications.

V. DIGITAL IMAGE PROCESSING

Digital image processing involves the manipulation and interpretation of digital images with the aid of a computer (Lillesand and Kiefer 1987). Digital image processing is an extremely broad subject and often involves procedures that can be mathematically complex, but the central idea behind digital image processing is quite simple. The digital image is fed into a computer one pixel at a time. The computer is programmed to insert these data into an equation, or series of equations, and then store the results of the computation for each pixel. These results form a new digital image that may be displayed or recorded in pictorial format or may itself be further manipulated by additional programs. The possible forms of digital image manipulation are literally infinite. However, virtually all these procedures may be categorized into one (or more) of the following four broad types of computer-assisted operations.

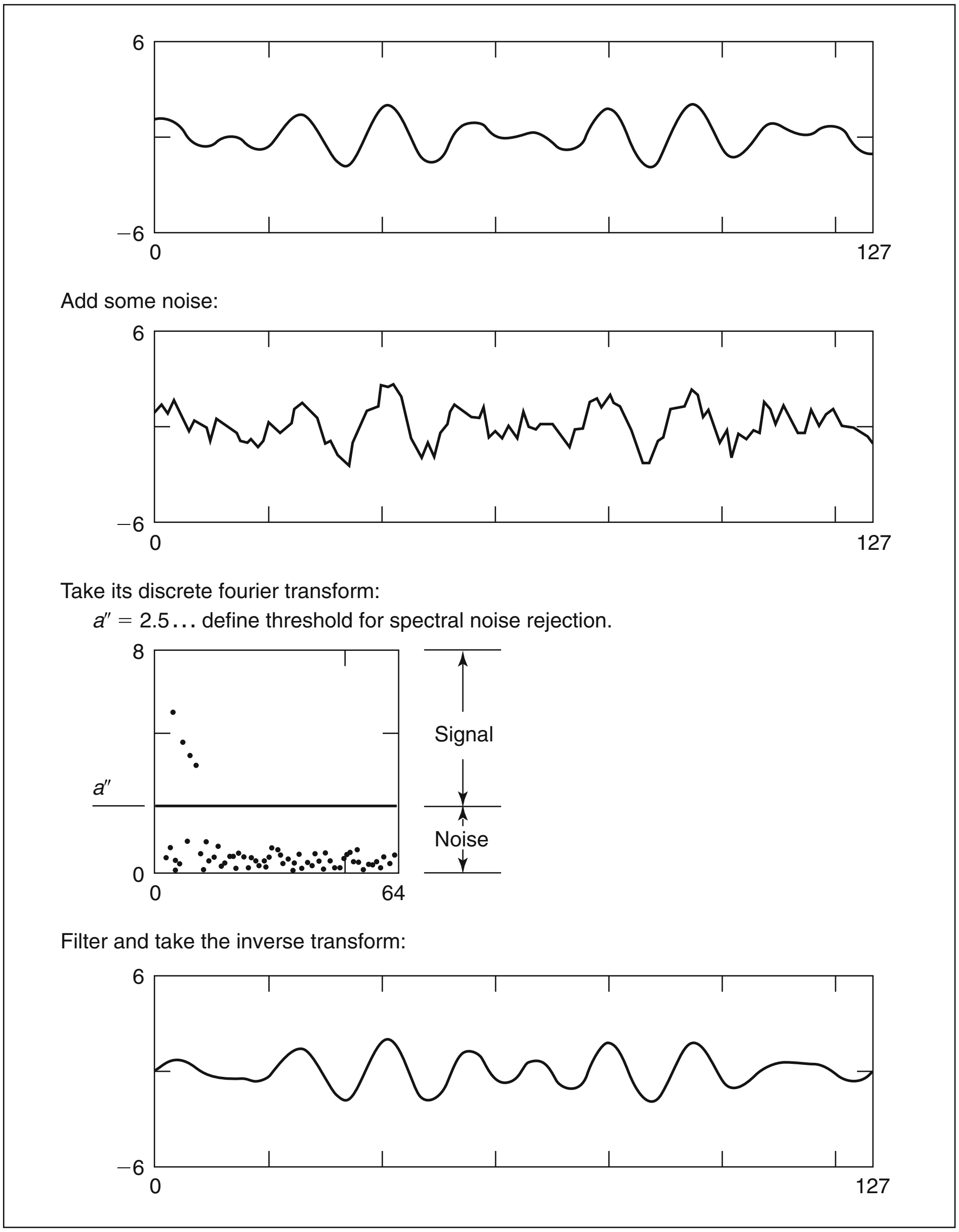
A. Image Rectification and Restoration

These operations aim to correct distorted or degraded image data to create a more faithful representation of the original scene. This typically involves the initial processing of raw image data to correct for geometric distortions, to calibrate the data radiometrically, and to eliminate noise present in the data. Geometric distortion can be induced into the image by sensor operation, orbital geometry, and earth geometry. Examples include the altitude, latitude, and velocity of the platform, earth curvature, atmospheric refraction, and non-linearities in the sensor field of view. All these contribute to distortion. These distortions can be systematic or random. Several techniques exist to correct for geometric distortions. Radiometric correction can be used to address problems caused by scene illumination, atmospheric conditions, viewing geometry, and instrument response. Sun elevation correction can account for seasonal position of the sun relative to the earth. Noise removal can be used to correct striping, boundary, and non-systematic variations that cause the images to be snowy. These can be removed by using a 3×3 or 5×5 median or averaging filter. Thus the nature of any particular image restoration process is highly dependent upon the characteristics of the sensor used to acquire the image data. Image rectification and restoration procedures are often termed preprocessing operations, because they normally precede further manipulation and analysis of the image data to extract specific information.

Fourier Transform Example

This example illustrates how noise can be removed spectrally from an image signal. First, one must remember that any signal can be represented as a combination of sine and cosine waves with different frequency[5], amplitude[6], and

Figure 6.15 FILTERING A NOISY SIGNAL WITH FOURIER TRANSFORM



Add some noise:

Take its discrete fourier transform:

a'' = 2.5... define threshold for spectral noise rejection.

Filter and take the inverse transform:

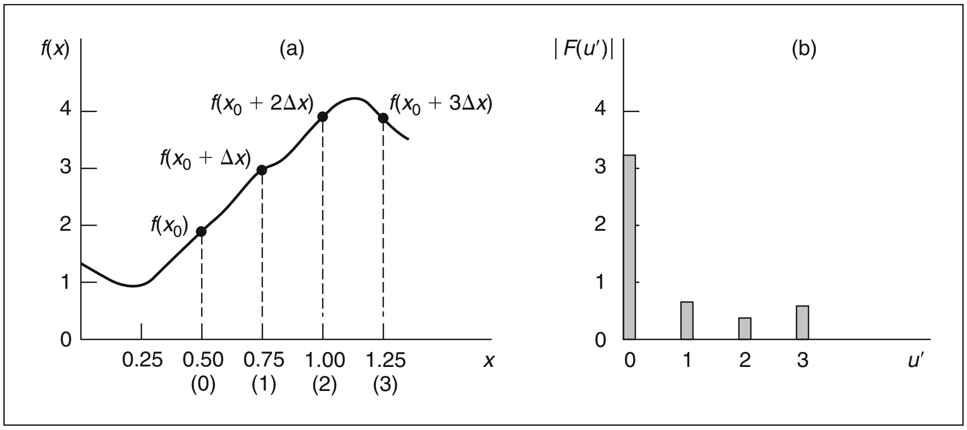
phase[7]. One common noise removing technique is the **Fourier transform,** which converts the signal into its frequency domain or into harmonics when one thinks of the voice signal (Gonzalez and Woods 1992). Inasmuch as noise has a very distinctly different harmonic compared with the regular signal (again think of the voice analogy), it can be recognized and easily removed in the harmonic or the frequency domain. An example would drive this point home. Shown in Figure 6.15 is a made-up signal, to which noise has been added. A Fourier transform has been taken of the signal (with its noise.) It can be seen that the noise has a harmonic quite a bit different from the signal, most of which are at the lower part of the frequency plot in the third frame of the figure. A frequency threshold of 2.5 in this case would form a very clear watershed between the signal and noise. Now we can remove, or "filter" out, any "signal" associated with the frequency below the threshold. This is commonly known as a high-pass noise-filter. An inverse transform is then taken, which reconstitutes the original signal with the noise removed. ■

1. Discrete Fourier Transform. More formally, let $f(x)$ be a continuous function of a real variable x. The Fourier transform of $f(x)$, denoted by $F(f(x))$, is defined as $F(f(x)) = F(u') = \int_{-\infty}^{\infty} f(x) \exp(-j2\pi u'x)\, dx$ where $j = \sqrt{-1}$, remembering that sine and cosine curves can be represented by a complex exponential function of frequency u'. Amplitude, or the Fourier spectrum, in this case is $|F(u')|$. Given $F(u')$, $f(x)$ can be recovered by using the inverse Fourier transform $F^{-1}(u') = f(x) = \int_{-\infty}^{\infty} F(u') \exp(j2\pi u'x)\, du'$. Suppose a continuous function $f(x)$ is now discretized into a sequence $\{f(x_0), f(x_0+\Delta x), f(x_0 + 2\Delta x), \ldots, f(x_0 + (n-1)\Delta x)\}$ by taking n samples Δx apart, as shown in Figure 6.16. The sequence $\{f(0), f(1), f(2), \ldots, f(n-1)\}$ now denotes any n uniformly spaced samples from the corresponding continuous function. The discrete Fourier transform (DFT) pair that applies to the sample functions is then given by

$$\begin{aligned} F(u') &= \frac{1}{n} \sum_{x=0}^{n-1} f(x) \exp(-j2\pi u'x/n) \\ f(x) = F^{-1}(u') &= \sum_{u'=0}^{n-1} F(u') \exp(j\, 2\, u'\, x/n) \end{aligned} \tag{6.1}$$

for $x = 0, 1, 2, \ldots, n-1$.

Figure 6.16 DISCRETE FOURIER TRANSFORM EXAMPLE



Application of this equation pair to the signal in Figure 6.17 yields $F(0) = 1/4 \sum_{x=0}^{n-1} f(x)\exp(0) = 1/4\,[f(0) + f(1) + f(2) + f(3)] = 1/4(2 + 3 + 4 + 4) = 3.25$ and $F(1) = 1/4 \sum_{x=0}^{3} f(x)\exp(-j2\pi x/4) = 1/4(2e^{0} + 3e^{-j\pi/2} + 4e^{-j\pi} + 4e^{-j3\pi/2}) = 1/4(-2 + j)$, remembering Euler's formula $e^{j\theta} = cos\,\theta + j sin\,\theta$ in the last part of the calculation. Continuing with this procedure gives $F(2) = -1/4(1 + j0)$ and $F(3) = -1/4(2 + j)$. All values of $f(x)$ contribute to each of the four terms of the discrete Fourier transform (DFT). Conversely, all terms of the transform contribute in forming the inverse transform via Equation 6.1. The Fourier spectrum is obtained from the magnitude of each of the transform terms: $|F(0)| = 3.25$, $|F(1)| = [(2/4)^2 + (1/4)^2]^{1/2} = \sqrt{5}/4 = 0.56$, $|F(2)| = [(1/4)^2 + (0/4)^2]^{1/2} = 1/4 = 0.25$, and $|F(3)| = [(2/4)^2 + (1/4)^2]^{1/2} = \sqrt{5}/4 = 0.56$. The spectrum is illustrated in frame (b) of Figure 6.7.

2. Fast Fourier Transform. The number of complex multiplications and additions required to implement Equation 6.1 is proportional to n^2, square of the number of discrete intervals. That is, for each of the n values of u', expansion of the summation requires n complex multiplications of $f(x)$ by $\exp(-j2\pi u'x/n)$ and $n - 1$ additions of the results. Proper decomposition of Equation 6.1 can make the number of multiplication and addition operations proportional to $n\log_2 n$. The decomposition procedure is called the fast Fourier transform (FFT) algorithm. The reduction in proportionality from n^2 to $n\log_2 n$ operations represents a significant savings in computational effort. The FFT approach offers a considerable computational advantage over direct implementation of the Fourier transform, particularly when n is relatively large. For that reason, many real-world applications use FFT rather than conventional discrete transform, including the example shown in Figure 6.16.

The above noise-removal procedures, both regular transforms and FFT, were illustrated for a single dimensional case. It can be shown that the same idea can be generalized to a two-dimensional image. The transform now has two arguments instead of one $F(u_1', u_2')$, corresponding to the frequencies in both dimensions u_1' and u_2'. Page limitation prevents further development of the two-dimensional transform here. Readers are referred to Gonzalez and Woods (1992) for an in-depth treatment of the methodology. In accordance with the application flavor of this book, however, we implemented the FFT for image processing in the TS-IP (Training System/Image Processing): a software distributed with this book. The readers are invited to experiment the FFT routine with the image files supplied with the program.

3. Spatial Filter. Rather than developing the two-dimensional Fourier transform, we choose to introduce the concept of the **spatial filter.** It accomplishes a similar noise removal function, but it is based on an entirely different principle. No longer do we need to work in the frequency domain. Considering that noise shows up as outliers in their signal intensity—i.e., either too weak or too strong—spatial filters do their work directly in the signal domain. The **average filter** or the **median filter** are two examples of a spatial filter. Both simply smooth out the outliers, replacing each outlier with a streamlined pixel. While both accomplish the mainstreaming task, they yield different results, as we will demonstrate. Again, the reader is invited to experiment with the average and median filters implemented in TS-IP and verify the results from the following example.

Example

The following example deals with possible outliers in satellite spatial imagery as the result of noise, where several methods could be used to remove the offending point(s). Among these are 3 × 3 averaging, which compares first- and second-order neighbors, and 3 × 3 and 5 × 5 median filtering. Both work on the principle of replacing bad outlier data points with good ones. We use the 5 × 5 data cell below to answer the following questions with regard to the center point outlier.

31	33	41	44	48
32	39	44	42	45
43	40	92	40	40
46	43	41	42	42
43	44	43	41	42

(a) Using the averaging method, calculate the first- and second-order averages. Describe how you might set threshold in this case and which value you would use as a replacement.

The center-pixel gray value, $v(0)$, is given as 92. The average of first-order neighbors is $v(1) = (44 + 40 + 41 + 40)/4 = 41.25$. The average of second-order neighbors is $v(2) = (42 + 42 + 43 + 39)/4 = 41.50$. One would next want to calculate the absolute differences $|v(0) - v(1)|$, $|v(0) - v(2)|$, $|v(1) - v(2)|$ and if some threshold is overcome, substitute either $v(1)$ or $v(2)$ for $v(0)$. In this case

$$\begin{aligned}|v(0) - v(1)| &= |92 - 41.25| = 50.75\\ |v(0) - v(2)| &= |92 - 41.50| = 50.50\\ |v(1) - v(2)| &= |41.25 - 41.50| = 0.25\end{aligned}$$

One can select any value for the threshold. It depends on how much one wants to smooth the data. One way is to say that if $|v(0) - v(1)|$ or $|v(0) - v(2)|$ is greater than 50 percent of the center outlier 92, then we replace $v(0)$ with $v(1)$ or $v(2)$. In this case $(0.5)(92) = 46.00$. Since 50.75 is larger than the threshold, a replacement is in order, or $v(0)' = v(1) = 41.25$.

(b) Now calculate the 3 × 3 and 5 × 5 median estimates of the center point. Describe how you might set the threshold in this case and whether or not you would replace the point.

The 3 × 3 median, $m(3)$, is 42 from the 9 entries of the 3 × 3 neighborhood:

39 40 40 41 (42) 42 43 44 92

The absolute difference is $|v(0) - m(3)| = |92 - 42| = 50$. Using the same 50 percent threshold criterion as in (a), the new value for the center point is $v(0)' = 42$. The 5 × 5 median $m(5)$ is 42 again from the 25 entries of the 5 × 5 neighborhood:

31 32 33 39 40 40 40 41 41 41 42 42
(42) 42 43 43 43 43 44 44 44 45 46 92

$|v(0) - m(5)| = |92 - 42| = 50$. Again using the same 50-percent threshold criterion as in (a), $v(0)' = 42$.

(c) What advantage does the median approach have over the averaging approach?

A median approach usually produces better results than an average approach because the average approach usually has a much higher replacement value than the median when dealing with noise. The median minimizes the effect of outliers, since they are not weighted more heavily. As a result, the median method does not cause blurring associated with averaging. This works better for noise that consists of spikes, and consequently, it is better in preserving edge sharpness. ■

Another way of smoothing spatial data is interpolation. Aside from simple interpolation, a family of techniques called **kriging** has been developed, which is designed to minimize the errors in the estimated values (Star and Estes 1990; Cressie 1991). The method is based on estimating the strength of the correlations between known data points, as a function of the distance between the points. This information is then used to select an optimal set of weights for the interpolation. Variations have been developed to include a trend surface model component, which permits estimating values outside the area of known points; this is not possible with simple distance weighted models.[8] Examples of this concept can be found in the "Ratio and correlation method" subsection of Chapter 2 when the time dimension is interpreted as the spatial dimension.

B. Image Enhancement

Enhancement procedures are applied to image data in order to more effectively display or record the data for subsequent visual interpretation. Normally, image enhancement involves techniques for increasing the visual distinction between features in a scene. The objective is to create new images from the original image data in order to increase the amount of information that can be visually interpreted from the data. The enhanced images can be displayed interactively on a monitor or they can be recorded in a hard copy format, either in black and white or in color. There are no simple rules for producing the single best image for a particular application. Often several enhancements made from the same raw image are necessary. Image enhancement can be accomplished by adjusting the contrast or doing spatial feature manipulation and multi-image manipulation. One can also zoom or enhance the resolution of this image. Enhancements involving multiple spectral bands of imagery can be made as well.

Convolution filters can be used for many purposes in image processing. Among them is edge detection, which again can be thought of as an image enhancement technique since it makes an image more crisp. Perhaps the best filters for this purpose come from the family known as Sobel operators, which are a kind of gradient operator to detect discontinuities (Gonzalez and Woods 1992). The gradient of an image $f(x, y)$ at location (x, y) is the vector $\nabla f(x, y) = (G_x, G_y)^T = (\partial f/\partial x, \partial f/\partial y)^T$. The simple gradient δf is the scalar $\delta f = |\nabla f(x, y)| = (G_x^2, + G_y^2)^{1/2}$. A common practice is to approximate the gradient with absolute values $\delta f = |G_x| + |G_y|$. The direction of the gradient vector is an important quantity; it shows whether we are moving from the left to the right or from the right to the left on the x-axis. Likewise, it shows whether we are moving up or down on the y-axis.

At this point, we need to formally define a **mask,** or its alternate names filter, window, or template. A mask is a "window" overlaid on top of an image with a set of specific mathematical operations performed on the pixels underneath this window. The idea behind mask operations is to let the value assigned to a subject pixel be a function of its gray level and the gray level of its neighbors. We have already seen this in the average filter in the last section, where an outlier is replaced by the average of its four first-order neighbors. The first-order neighbors in this case form the 4-element mask for the subject pixel, consisting of a weight of 1/4 each. In other words, the mask looks like

$$\begin{bmatrix} 0 & 1/4 & 0 \\ 1/4 & 0 & 1/4 \\ 0 & 1/4 & 0 \end{bmatrix}$$

Another example can be found in the median filter example, where the outlier is replaced by the median of the 3 × 3 window (nine pixels) centering around the subject outlier pixel. Instead of a weighted average, the operator is now median computation. In image processing, a mask is normally applied like a moving window across an image, centering around each and every pixel in the image until all pixels have been visited. The result is a processed image with either noise removal or image enhancement accomplished. The Sobel mask can be thought of as a combination of a differencing mask of the weights $(-1\ 0\ 1)$ (or its vertical counterpart) followed by a smoothing mask of the weights (1 2 1) (or its vertical counterpart), as we will show immediately below. The differencing mask accentuates the gray-value differences among first-order neighbors, while the smoothing mask averages the subject pixel and its first-order neighbors. Because derivatives enhance noise, the smoothing effect is a particularly attractive feature. Given the 3 × 3 Sobel operators $G_{x\uparrow}$, $G_{y\rightarrow}$, $G_{x\downarrow}$, and $G_{y\leftarrow}$

$$G_{x\downarrow} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \qquad G_{y\rightarrow} = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \tag{6.2}$$

$$G_{x\uparrow} = \begin{bmatrix} 1 & 2 & -1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \qquad G_{y\leftarrow} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} \tag{6.3}$$

and the data cell

$$\begin{bmatrix} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{bmatrix}$$

derivatives based on the Sobel operator masks are

$$\begin{aligned} |G_x| &= |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| \\ |G_y| &= |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)| \end{aligned} \tag{6.4}$$

Computation of the gradient at the location of the center of the masks can be performed with these equations, giving one value of the gradient δf. To get the next value, the masks are moved to the next pixel location and the procedure is repeated.

Example
Using the sample 3 × 3 data-cell

$$\begin{bmatrix} 85 & 112 & 150 \\ 82 & 63 & 115 \\ 84 & 80 & 127 \end{bmatrix}$$

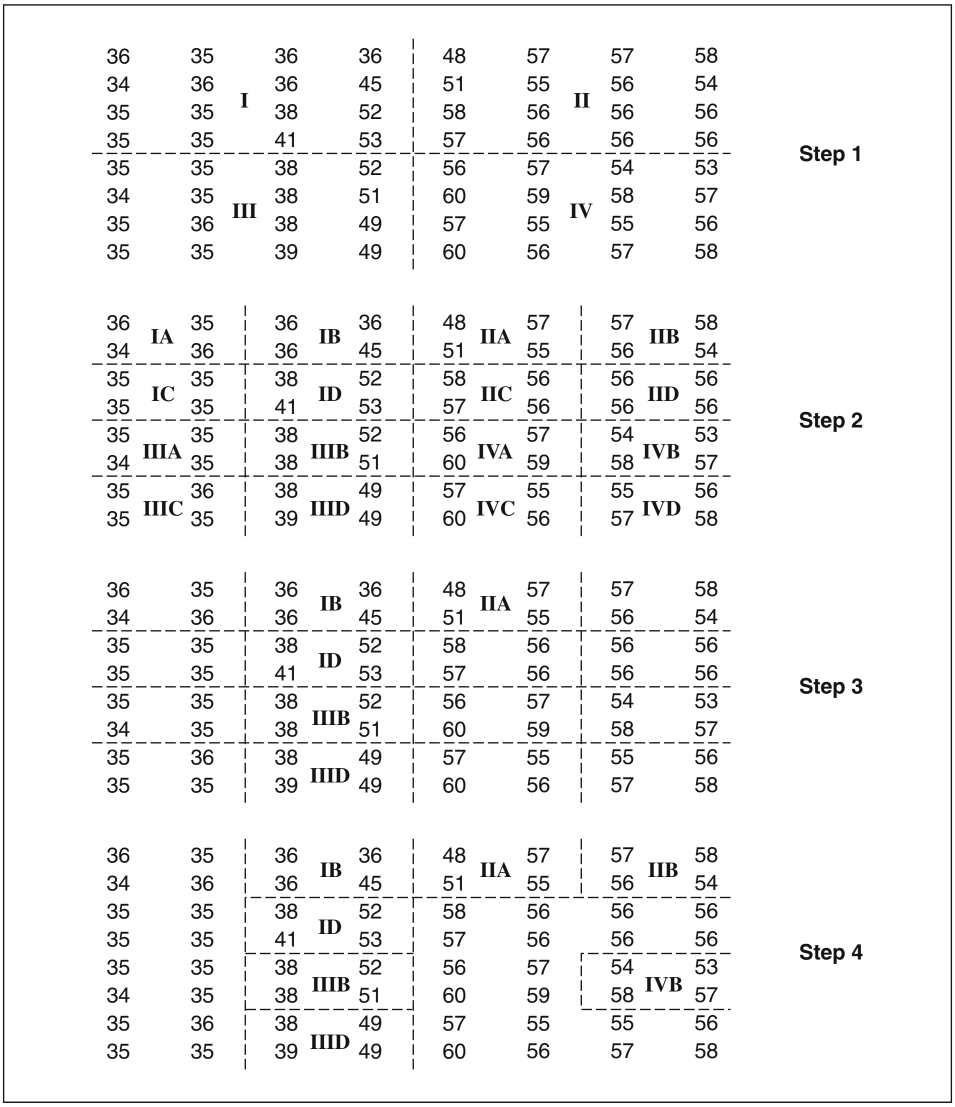
which of the four Sobel filters, $G_{x\uparrow}$, $G_{y\rightarrow}$, $G_{x\downarrow}$, and $G_{y\leftarrow}$, do you think would most likely detect the edge as part of a change in the data pattern? What is the resulting convolved value, or the gradient, for the center pixel using this filter? Visual inspection of the data indicates the brightest line is the third column $(150\ \ 115\ \ 127)^T$. The x-axis in G_x is defined in the vertical direction so the strongest response produced by $|G_x|$ is an edge parallel to the x-axis. This data set seems to have an edge perpendicular to the x-axis. So we should use a G_y operator. One can use either the $G_{y\rightarrow}$ or $G_{y\leftarrow}$ since oftentimes, only absolute values are of interest, as shown in Equation 6.4. Using $G_{y\rightarrow}$ of Equation 6.3, it produces a value of $150 + (2)(115) + 127 - 85 - (2)(82) - 84 = 174$ for the center point. The fact that this value is much greater than the values from the $|G_x|$ operators (88) shows that the gradient is working best in the horizontal direction in detecting the edge $(150\ \ 115\ \ 127)^T$. ■

C. Image Classification

The objective of classification operations is to replace visual analysis of the image data with quantitative techniques for automating the identification of features in a scene. This normally involves the analysis of multispectral image data and the application of statistically based decision rules for determining the land cover identity of each pixel in an image. One can perform image classification that will categorize all pixels in an image into land cover classes like grass, water, sand, and so forth. When these decision rules are based solely on the spectral radiances observed in the data, we refer to the classification process as **spectral pattern recognition**. In contrast, the decision rules may be based on the geometrical shapes, sizes, and patterns present in the image data. These procedures fall into the domain of **spatial pattern recognition**. In either case, the intent of the classification process is to categorize all pixels in a digital image into one of several land cover classes or themes. These categorized data may then be used to produce thematic maps of the land cover present in an image, and/or to produce summary statistics on the areas covered by each land cover type.

In both spectral and spatial image classification, the problem can be viewed as grouping similar gray values together in two or more dimensional space. Consider the two-dimensional illustration in Figure 6.17, which can represent both a spectral or spatial image. In the latter case, the entries will simply be gray values in regular raster grid. In the former case, each cell represents a pair of coordinates (x, y), where x is a reading on one spectral band and y is the reading on the second band. Classification amounts to grouping pixels of similar gray values together in the former case or pixels with similar spectral

Figure 6.17 DISAGGREGATION AND AGGREGATION OF DIGITAL IMAGE



Step 1

36	35	36	36	48	57	57	58
34	36	36	45	51	55	56	54
35	35	38	52	58	56	56	56
35	35	41	53	57	56	56	56
35	35	38	52	56	57	54	53
34	35	38	51	60	59	58	57
35	36	38	49	57	55	55	56
35	35	39	49	60	56	57	58

Step 2

36	35	36	36	48	57	57	58
34	36	36	45	51	55	56	54
35	35	38	52	58	56	56	56
35	35	41	53	57	56	56	56
35	35	38	52	56	57	54	53
34	35	38	51	60	59	58	57
35	36	38	49	57	55	55	56
35	35	39	49	60	56	57	58

Step 3

36	35	36	36	48	57	57	58
34	36	36	45	51	55	56	54
35	35	38	52	58	56	56	56
35	35	41	53	57	56	56	56
35	35	38	52	56	57	54	53
34	35	38	51	60	59	58	57
35	36	38	49	57	55	55	56
35	35	39	49	60	56	57	58

Step 4

36	35	36	36	48	57	57	58
34	36	36	45	51	55	56	54
35	35	38	52	58	56	56	56
35	35	41	53	57	56	56	56
35	35	38	52	56	57	54	53
34	35	38	51	60	59	58	57
35	36	38	49	57	55	55	56
35	35	39	49	60	56	57	58

band readings in the latter case. In both cases, we classify image into the logical land cover types.

To illustrate the concept of classification, the region-oriented segmentation algorithm of Gonzalez and Woods (1992) may be of interest. First, a decision criterion describing the image is specified, such as the gray value range that describes, for example, a cornfield, a lake, or a beach. The digital image is then taken as a single region that is partitioned by repeated splitting. One method of dividing the image is by bisection. If the image does not meet the decision criteria, the image is divided into quadrants. If a quadrant does not meet the decision criteria, we divide it into subquadrants and so on. As the image is split into various sized regions, adjacent regions that meet the decision criteria can be merged. This splitting and merging continues until no further merging or splitting is possible. The end result is the objects of interest identified in the image.

In more formal terms, let R'' represent the entire image region. We may view segmentation as a process that spatially partitions R'' into n' subregions, R_1, $R_2, \ldots, R_n{}'$ such that (a) $\bigcup_{i=1}^{n'} R_i = R''$; (b) R_i is a connected region, $i = 1, 2, \ldots, n'$; (c) $R_i \cap R_j = \varnothing$ for all i and j, $i \neq j$; (d) $\mathbf{P}(R_i) =$ TRUE for $i = 1, 2, \ldots, n'$; and (e) $\mathbf{P}(R_i \cup R_j) =$ FALSE for $i \neq j$ where $\mathbf{P}(R_i)$ is a logical predicate over the points in set R_i such as the range of gray values. Specification regarding no overlaps between two subregions (Condition (c)) may be relaxed for multispectral classification, where the x-axis corresponds to one spectral band and the y-axis another. Notice this algorithm, while conceptually simple, is computationally explosive for any practical image. For this reason, it is good for fixing ideas only.

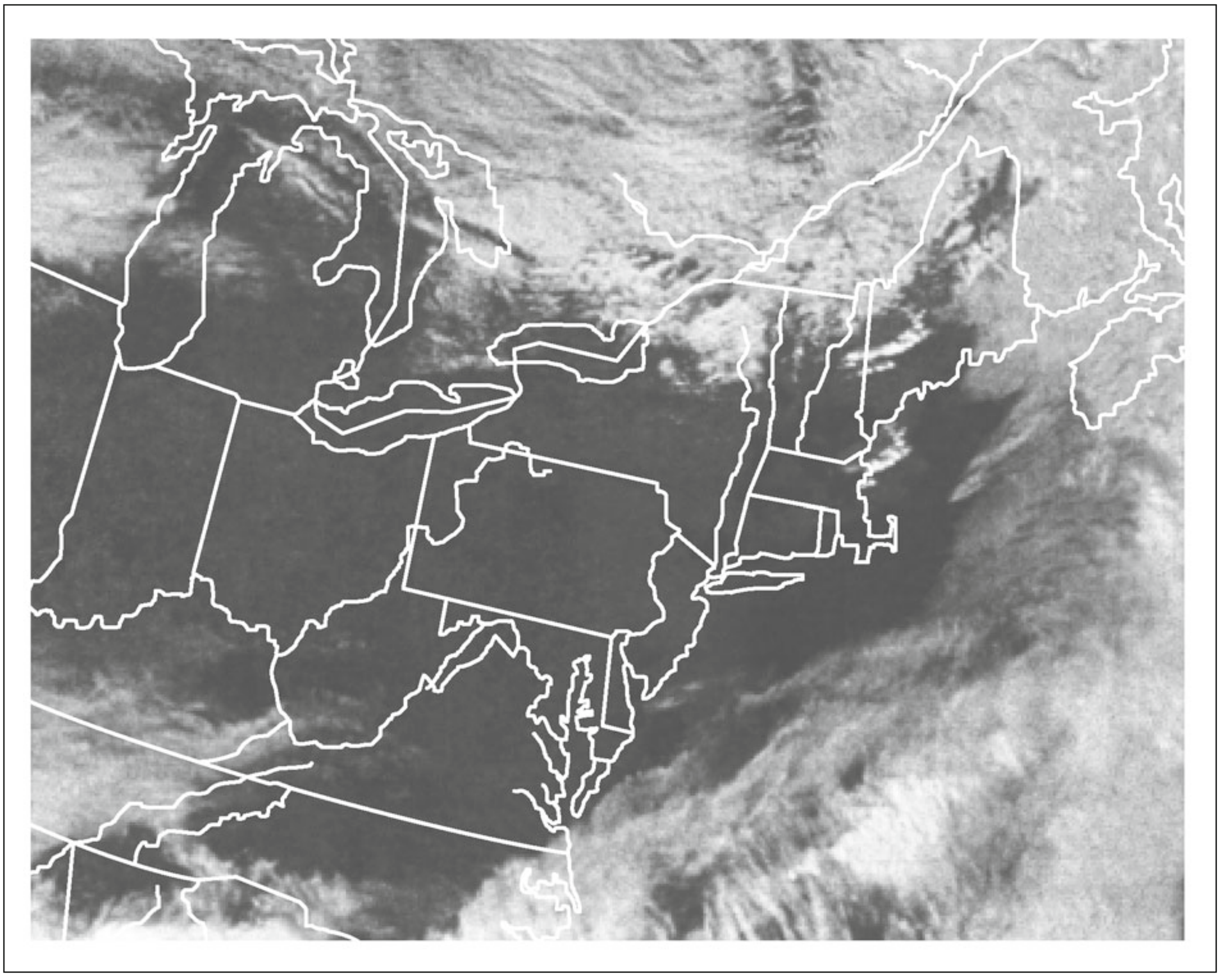
Example
Use the image cell in Figure 6.18 to demonstrate the concept of classification using the region splitting (disaggregation) technique. Do not split any cell smaller than 2 × 2 pixels. Show and briefly explain each step of the process. Re-aggregate cells, as necessary, at the end. Your predicate for each region is that the range of pixel gray values must be no greater than five. Be sure your final classification shows each separate region clearly and mark any region that fails to satisfy the predicate. (Note that this should not normally occur in practice).

First, we check the given region. Since $\mathbf{P}(R'') =$ FALSE (or the range of pixel gray values exceeds 5), we subdivide the region into four areas, labeled I, II, III and IV. This procedure was repeated for the second time, resulting in 16 areas, labeled IA, IB, IC, ID, IIA, IIB . . . and so forth (see steps 1 and 2 in Figure 6.18). At this point, we have the minimum 2 × 2 areas, which appear to satisfy the predicate, except for subregions IB, IIA, ID, IIIB, and IIID. Now aggregate the cells, checking to see that all cells assigned to a similar region have values that do not range more than 5 (as shown in step 3 of Figure 6.17). Subregions IA and IC can be recombined with IIIA and IIIC since they together satisfy the predicate. Subregions IIC and IID, together with IVA, IVC, and IVD can be combined, leaving IIB and IVB alone since their inclusion would violate the predicate (step 4). Notice the two subregions IIB and IVB which satisfy the predicate in the final partitions are not contiguous, but it is an acceptable answer as far as illustrating this algorithm is concerned. It is clear that this algorithm leads toward one particular classification and that other combinations are possible should a different partitioning algorithm be used. The classification of the same land cover can also be different should we shift the image by one pixel column to the right. ■

D. Data Merging

The next set of procedures in image processing is **data merging.** This procedure is used to combine image data for a given geographic area with other geographically referenced data sets for the same area. These other data sets might simply consist of image data generated on other dates by the same sensor, by other remote sensing systems, or an independently assembled data set. Frequently, the intent of data merging is to combine remotely sensed data with other resources of information into a GIS. For example, in urban applications image data are often combined with soil, topographic, ownership, zoning, and assessment

Figure 6.18 OVERLAY OF JURISDICTIONAL AND NATURAL BOUNDARIES ON WEATHER INFORMATION



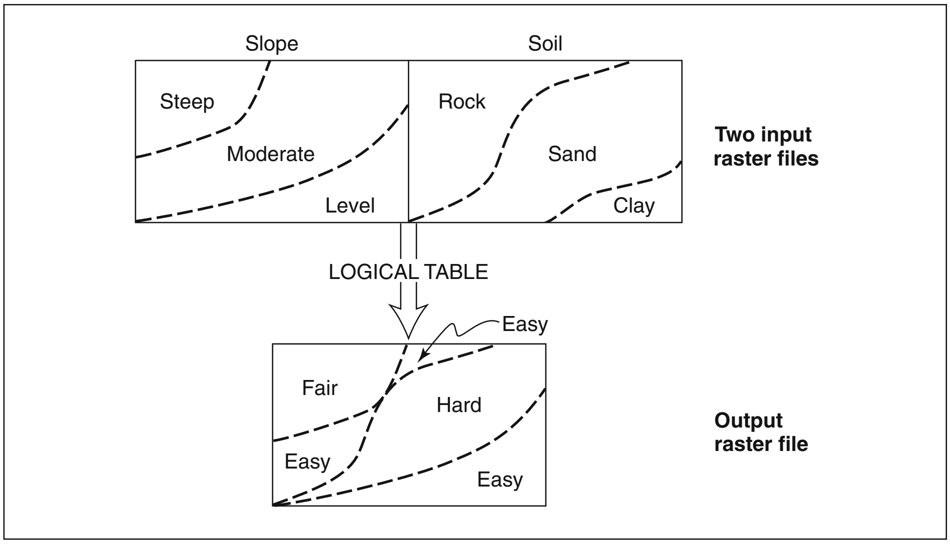
SOURCE: Courtesy of T.S. Kelso. Reprinted with permission.

information. This forms the basis for a GIS. A simple application of data merging is found in weather forecasts. A weather satellite such as ones launched by NOAA can provide a convenient image of cloud cover and other geographic information. But such information is often of little use unless it is referenced against jurisdictional boundaries such as nations, states, counties, and the like. Figure 6.18 shows the overlay of national, state boundaries, coastlines, and the Great Lakes on a NOAA satellite image of the northeastern United States and east Canada. Such a picture represents the merging of two data layers, one from the satellite and another from an archive storage. It is generated from the TS-IP software that comes with this book.

Example

Another application will show the potential of the data merge function much better (Star and Estes 1990). It is a trafficability problem that addresses the question: "Can a vehicle travel across a terrain with a certain slope and type of soil?"

Figure 6.19 A TRAFFICABILITY EXAMPLE OF DATA MERGING



SOURCE: Star and Estes (1990). Reprinted with permission.

Figure 6.19 shows the slope and soil data layers which form the information base for merging. Table 6.3 represents supplementary information regarding how easy it is for the vehicle in question to navigate a particular combination of slope and surface soil type. For example, when the slope is moderate, the vehicle cannot travel over sandy soil, but can travel on a gravel surface. This logical table forms a third data layer, consisting of the translation of nominal information (such as the rock/sand/clay categories) and ordinal information (such as level/moderate/steep as well as easy/fair/hard) distinctions into trafficability. It contrasts with scaled, or interval information typically found in raster files. This layer is input to the data merging process on top of the two data layers on soil and slope. The resulting derived suitability map for traversal, or the output of the entire exercise, is obtained at the bottom of Figure 6.19.

In a raster-based system, each cell in the input data provides a soil/slope data-tuples input to the trafficability table, which in turn determines the class of trafficability in the output data. We read the value of the first element in the soil array and the first element in the slope array, send these

Table 6.3 INPUT DATA LAYER REGARDING TRAFFICABILITY

		Soil type		
		Rock	**Sand**	**Clay**
Slope	**Level**	Easy	Easy	Easy
	Moderate	Easy	Hard	Fair
	Steep	Fair	Hard	Hard

SOURCE: Star and Estes (1990). Reprinted with permission.

values to a routine that derives the resulting trafficability class, and send this derived value to the first element in the output array. This process continues through all the elements in the raster arrays. While the data merging algorithm is fairly straightforward with raster files, the process is much more involved in vector based files, although the data storage requirement is much more compact in comparison. ■

Although the above treats the four procedures of digital image processing—rectification, enhancement, classification, and merge—as distinct operations, they all interrelate. For example, the restoration process of noise removal can often be considered an enhancement procedure. Likewise, certain enhancement procedures can be used not only to enhance the data, but also to improve the efficiency of classification operations. In a similar vein, data merging can be used in image classification in order to improve classification accuracy as in the combination of multispectral bands. In this regard, we will conduct a case study of merging multispectral bands later on in Section XI. Hence the boundaries between the various operations we discuss separately here are not well-defined in practice.

VI. DIGITAL IMAGE PROCESSING SOFTWARE AND HARDWARE

Image processing is, in general, a special form of two-dimensional, and sometimes three-dimensional, signal processing of scenes collected by sensors. Digital data of these scenes are stored on computers in bits. One bit of information is either on (1) or off (0). If an image had 1 bit of information on it, there will be two gray levels in it: white and black. As more bits of data are added, the number of gray levels in the picture increases. With 6 bits of data, there are 2^6 or 64 gray levels, ranging in value from 0 (black) up to 63 (white). The number of bits in a given pixel determines the number of unique gray values (or colors) available. Eight-bit pixels, for example, provide 256 different gray values in white to black shades or 256 unique colors in a pseudocolor mode.

Computer systems for image processing range from microcomputers to mainframe. Dedicated image processing systems include display memory, a video processor, a parallel interface to a computer, a human-machine interface, digital-analog converters, and a comprehensive software subroutine library. The basic subroutine library should contain all the necessary software for manipulating the internal parts of the image processor. A frame buffer is the key to any image processing system. This bank of memory stores the image data. Most medium size systems are several banks of 512×512 elements. The rows of the frame buffer matrix are the lines of the image, and the columns along each line are the samples. A digital-analog (D-A) converter transforms the contents of the image memory into a form compatible with the monitor. The number of different intensity-levels that a D-A converter can output is related to the number of bits it is designed to handle; the more bits, the more distinct colors or gray levels it can produce. An important part of an image processing system is a look-up table, which is a table of stored data for reference purposes. The look-up table performs a transformation or mapping between each unique input data value and some predefined output values. Table 6.3 represents a more sophisticated example of such a look-up table.

An instructional image processing software is included on the CD/DVD at the back of this book. The TS-IP software (Kelso et al. 1995) runs under Microsoft Windows or MS-DOS on an appropriate Personal Computer (PC) under a 256 VGA graphics card and a VGA monitor. Several resolution options are available depending on the specific PC, including 640 × 400 and 640 × 480. Among the features offered by TS-IP are:

(a) adding an image to the image in the current window, resulting in an overlay (an example is shown in Figure 6.19);

(b) examining an image by viewing a pixel located at an x-y coordinate and displaying its gray value, or viewing a line of pixels and displaying the gray values along the line; when combined with the operation described in (c) below, this allows for image restoration and enhancement;

(c) displaying a histogram of the number of pixels in an image by gray values, which allows for the truncation of the low and/or high gray value range such as that associated with high-level clouds (allowing one to "see through the clouds");

(d) restoring or enhancing an image via such filters as Sobel convolution[9],fast Fourier transform[10], and median filter[11];

(e) performing contour plots of an image where the contours correspond to a specified gray value;

(f) highlighting the image with desired color scheme, including colors of the rainbow or simply a 256 gray value scale.

Real-life satellite images can be handled within TS-IP. The size of the image is limited mainly by the secondary storage device available for filing these images and the display memory. A bank of public domain satellite images is included on the CD/DVD that accompanies this text. Instead of being a production line software, TS-IP is mainly intended to demonstrate the power of image processing as described in Section VI, including image rectification and restoration, image enhancement, image classification, and data merging. Through these image processing functions, one can show such interesting features as the capacity to see through clouds. This is achievable through a combination of feature (c) and stretching the remaining gray values to fill in the upper range vacated by the removal of high-level clouds. While the restoration and enhancement functions are accomplished well, the classification feature is yet to be implemented at this time.

VII. APPLICATIONS OF REMOTE SENSING

It is clear that remote sensing devices have facilitated a fair amount of planning applications. For example, there are documented evidences of its usefulness in environmental, land use, and hazard mitigation studies (Sabins 1987). NOAA satellites, for example, use the advanced very high resolution radiometer (AVHRR), a cross-track multispectral scanner that acquires images with an image swath width of 2700 km (1768 mi) and a ground resolution cell of 1.1 by 1.1 km (0.634 mi). Table 6.4 shows the spectral bands of AVHRR. Spectral ranges of AVHRR bands 1 and 2 were positioned to record significant vegetation properties. As shown by the vegetation reflectance curve in Figure 6.20, the readings in band 1 (B_1) records the chlorophyll absorption of red wavelengths. Band 2 (B_2) records the strong reflection of infrared (IR) wavelengths by the cell structure of leaves. The ratio B_2/B_1 is one index of vegetation. Another is the spectral or normalized vegetation index (NVI), a relationship defined as

$$NVI = \frac{B_2 - B_1}{B_2 + B_1} \tag{6.5}$$

This ratio is more useful than individual bands because it brings out the contrast and largely eliminates reflectance variation due to differences in solar elevation. The values for B_1 and B_2 are the average values for the reflectance curves at those wavelength intervals. For the vegetation spectrum in Figure 6.20, the spectral vegetation index is calculated as 0.41. For the dry soil spectrum, the index is only 0.30. Various proportions of soil and vegetation in a ground resolution cell of an AVHRR image will result in intermediate values. Also, different types of vegetation and soil may have different index values from those in Figure 6.20. Individual NVI maps can be used to prepare a vegetation classification map in color codes.

AVHRR images are well suited for studying vegetation distribution and seasonal changes in a continent-wide scale for the following reasons:

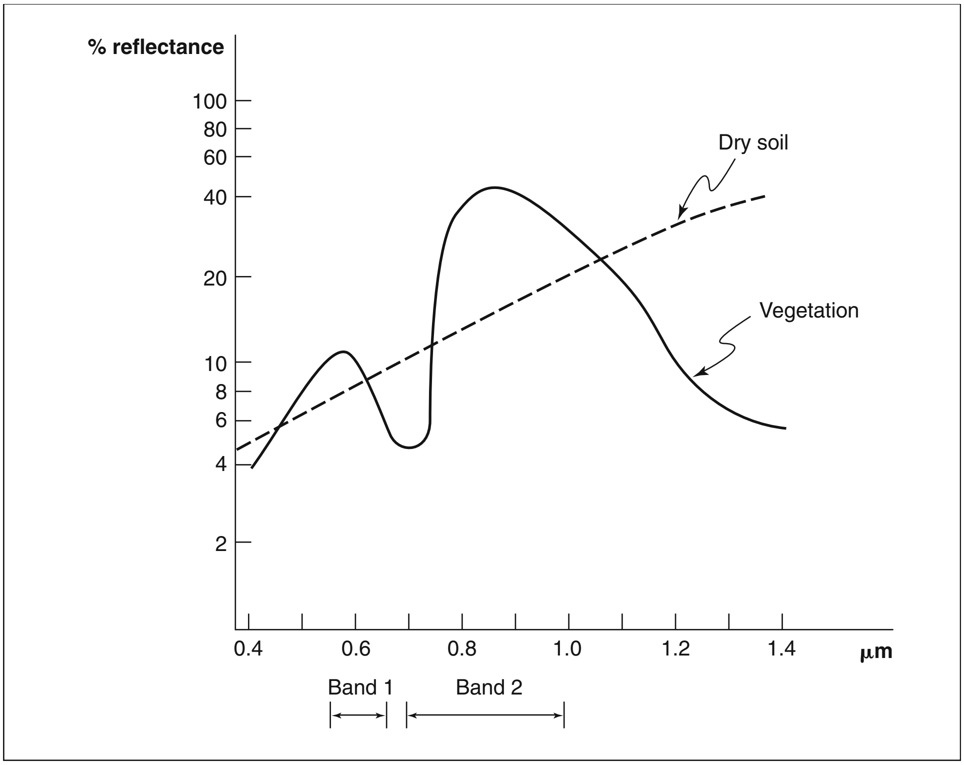
(a) The 2700-km (1768 mi) image swath of AVHRR covers a continent such as Africa with a few images, while 1100 LANDSAT MSS (multispectral scanner) or TM (thematic mapper) images are required.

Table 6.4 REMOTE SENSING CHARACTERISTICS OF THE ADVANCED VERY HIGH RESOLUTION RADIOMETER

Band	Wavelength, μm	Remarks
1	0.55–0.68	Red: for daytime clouds and vegetation
2	0.73–1.10	Reflected IR: for shorelines and vegetation
3	3.55–3.93	Thermal IR: for hot targets such as fires and volcanoes
4	10.50–11.50	Thermal IR: for sea temperatures and for daytime and nighttime clouds
5	11.50–12.50	Thermal IR: recorded only on NOAA 7 satellites & beyond

SOURCE: Sabins (1987). Reprinted with permission.

Figure 6.20 REFLECTANCE SPECTRA OF VEGETATION AND DRY SOIL



SOURCE: Sabins (1987). Reprinted with permission.

(b) The daily repetition of AVHRR provides a wide selection of images for seasonal changes and for cloud-free coverage. By contrast, LANDSAT TM operates on a 16-day repetition cycle.

(c) The 4-km (2.49 mi) pixels of AVHRR are adequate for regional studies, while the 79-m (259 ft) or 30-m (98.36 ft) pixels of MSS or TM result in far too much data for economical processing.

Another application of remote sensing information is in urban land use. Utilizing the six TM bands of visible and reflected infrared (IR) data, classification of land use can be performed, in much the same manner as the NOAA satellite discussed above. For example, classification may be color coded as follows: violet (residential), orange (commercial), black (streets and parking lots), gray (construction sites), blue (open land), dark green (irrigated vegetation), medium green (mixed rangeland), light green (shrub and brushland), yellow (sand and gravel).

Urban areas are so diverse in their land use that even the higher resolution of LANDSAT TM may not be able to represent these diversities adequately. A typical suburban residential lot is approximately the size of one TM 30-by-30-m (98.36-by-98.36 ft) ground resolution cell. The lot will include some or all of the following materials: trees and shrubs, lawns, paving and sidewalks, roofs, and water for a swimming pool. For such a cell, the digital numbers of the TM bands for that pixel are a composite of the spectral reflectance of the various materials. Despite these problems, the LANDSAT classification map portrays quite well the major categories of land use and land cover. Obviously, a multilevel imaging scheme is preferred, ranging from LANDSAT MSS images to low-altitude aerial photographs. Table 6.5 tabulates the spectrum of scale resolutions obtainable from each remote sensing device.

LANDSAT MSS images are excellent for recognizing the continuity and regional relationships of faults. The higher spatial resolution of LANDSAT RBV (return beam vidicon) and TM images records many of the topographic features indicative of active faulting. Stereo viewing of aircraft and large format camera (LFC) photographs provides detailed information on geomorphic features formed by faulting. Thermal IR images of arid and semi-arid areas may record the presence of active faults with little or no surface expression, such as the San Andreas and Superstition Hills faults. The highlighting and shadowing effects on low sun angle aerial photographs can emphasize topographic scarps associated with active faults, such as in the Carson Range, Nevada. Radar images also emphasize subtle features along active fault zones, such as shown in the Spaceborne Imaging Radar, SIR-A, image of the Superstition fault in Iranian Jaya, Indonesia. After a hiatus of nearly 10 years, the most sophisticated imaging radar ever flown in space was launched in 1994 (Shen 1995). Both the Spaceborne Imaging Radar (SIR-C) and X-band Synthetic Aperture Radar (X-SAR) operate in the microwave regime and are able to generate high-resolution images immune to blockage or perturbation from micro particles such as clouds and rain. Like most radar systems, both systems provide their own illumination, enabling 24-hour operation.

Analysis of remote sensing information is directly tied to GIS technology. Infrared images when included in a GIS analysis can reveal considerable information about land use, vegetation growth, and environmental problems. Digitized remotely sensed images can easily become a layer against which other database can be compared, as suggested previously. Given the potential for high-resolution satellite imagery to supplement traditional ground-based data, McCord et al. (1996) estimated the daily highway coverage that could be obtained from a sensor carried on an orbiting satellite. It was found that if a satellite orbit were designed to maximize traffic monitoring coverage, approximately 0.4 percent of the continental U.S. could be imaged daily at 1-meter resolution, a resolution that should be sufficient to distinguish trucks from passenger cars. For orbital inclination angles more typical of earth

Table 6.5 MULTILEVEL CLASSIFICATION OF IMAGES

Level	System	Image Scale
I	Landsat MSS images	1:250,000 and smaller
II	Landsat TM images and high-altitude aerial photographs	1:80,000 and smaller
III	Medium altitude aerial photographs	1:20,000 to 1:80,000
IV	Low altitude aerial photographs	Larger than 1:20,000

SOURCE: Anderson et al. (1976). Reprinted with permission.

observation satellites, the coverage drops to approximately 0.2 percent. Coverage could be increased markedly with improved image processing and interpretation and data compression algorithms.

Global navigation satellite systems (GNSS) employ 24 satellites to determine three-dimensioned geocentric positions by distance measurements. They include both the U.S. Global Positioning System (GPS) and the GLONASS of the Commonwealth of Independent States. The accuracy of positions determined by GNSS is highly variable depending on the mode employed. A single receiver only provides geodetic positions with an accuracy of about 100 meters (333 ft). With two receivers, one can make use of the differential mode, which yields accuracies of about 1–5 meters (3.3 to 16.7 ft). Most importantly, real time location information of GNSS can be relayed back to a central GIS from a service vehicle in the field, allowing for optimal routing of the vehicle, as is being practiced in Intelligent Transportation Systems (electronic highways). Artificial intelligence programs are being used to assist in data entry, map interpretation, and information retrieval. A spatial data infrastructure can eventually be accessible from the Internet or information superhighway. This would serve as an electronic index of the available geographic databases to anyone with a personal computer. The results is to avoid duplication of effort by knowing what data has already been compiled.

VIII. SPECTRAL VERSUS SPATIAL PATTERN RECOGNITION

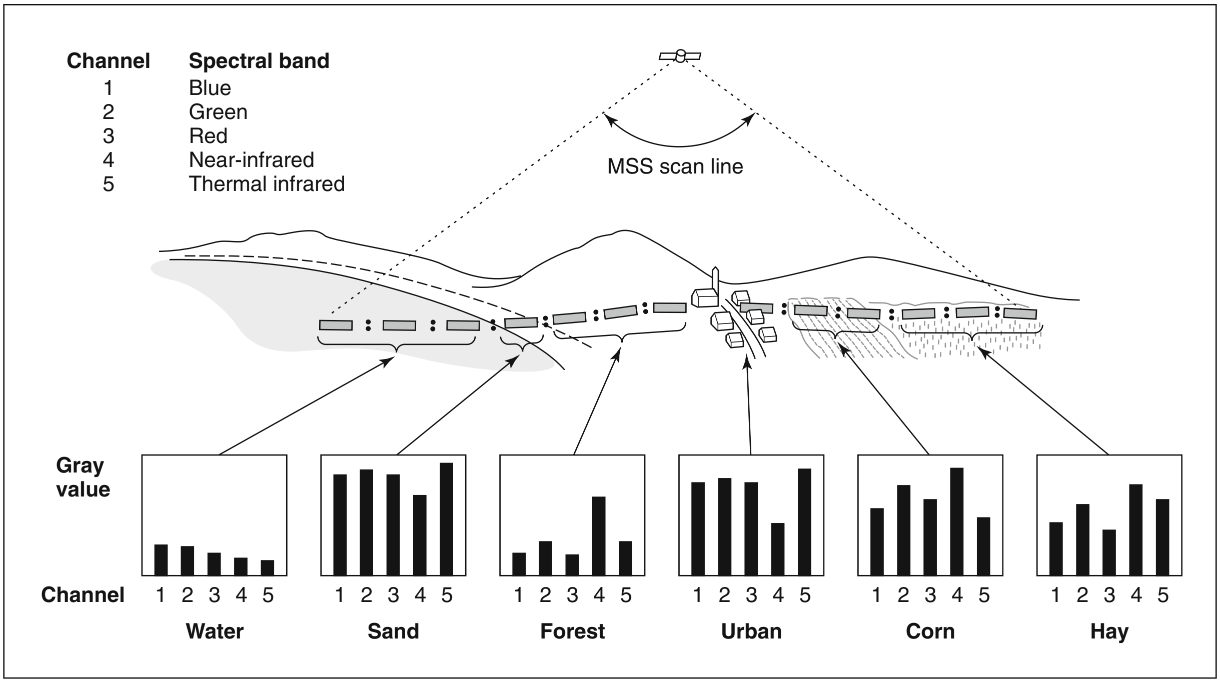
As mentioned previously, the overall objective of image classification is to categorize all pixels in an image into land cover classes or themes. Normally, multispectral data are used to perform the classification and, indeed, the spectral pattern present within the data for each pixel is used as the numerical basis for categorization. That is, different feature types manifest different combinations of digital numbers (DNs) based on their inherent spectral reflectance and emittance properties. In this light, a spectral pattern is not at all geometric in character. Rather, the term pattern refers to the set of radiance measurements obtained in the various wavelength bands for each pixel. As previously defined, spectral pattern recognition refers to the family of classification procedures that utilizes the pixel-by-pixel spectral information as the basis for automated land cover classification. Spatial pattern recognition, on the other hand, involves the categorization of image pixels on the basis of their spatial relationship with pixels surrounding them. Spatial classifiers might consider such aspects as image texture, pixel proximity, feature size, shape, directionality, repetition, and context. These types of classifiers attempt to replicate the kind of spatial synthesis done by the human analyst during the visual interpretation process. Accordingly, they tend to be much more complex and computationally intensive than spectral pattern recognition.

A. Spectral Pattern Recognition

Spectral pattern recognition forms the backbone of land cover mapping. Supervised classification refers to the process in which numerical description of the various land cover types present in a scene serves as an interpretation key that describes the spectral attributes for each feature type of interest. Each pixel in the data set is then compared numerically to each category in the interpretation key and labeled with the name of the category it most resembles. An example is taken from Lillesand and Kiefer (1987) to illustrate supervised classification. Figure 6.21 shows a single line of an airborne MSS data collected over a landscape composed of several cover types. For each of the pixels shown along this line, the MSS has measured scene radiance in terms of DNs recorded in each of the five spectral bands of sensing: blue, green, red, near-infrared, and thermal infrared. Below the scan line, typical DNs measured over six different land cover types are shown. The vertical bars indicate the relative gray values in each spectral band. These five outputs represent a coarse description of the spectral response patterns of the various terrain features along the scan line. If these spectral patterns are sufficiently distinct for each feature type, they may form the basis for image classification.

Figure 6.22 summarizes the three basic steps involved in a typical supervised classification procedure. In the training stage, the analyst identifies representative training areas and develops a numerical description of the spectral attributes of each land cover type of interest in the scene. Next, in the classification stage, each pixel in the image data set is categorized into the land-cover class it most closely resembles. If the pixel is insufficiently similar to any training data set, it is usually labeled unknown. The category label assigned to each pixel in this process is then recorded in the corresponding cell of an interpreted data set (an output image). Thus the multidimensional image matrix is used to develop a corresponding matrix of interpreted land-cover category

Figure 6.21 MEASUREMENTS MADE ALONG ONE SCAN LINE



SOURCE: Lillesand and Kiefer (1987). Reprinted with permission.

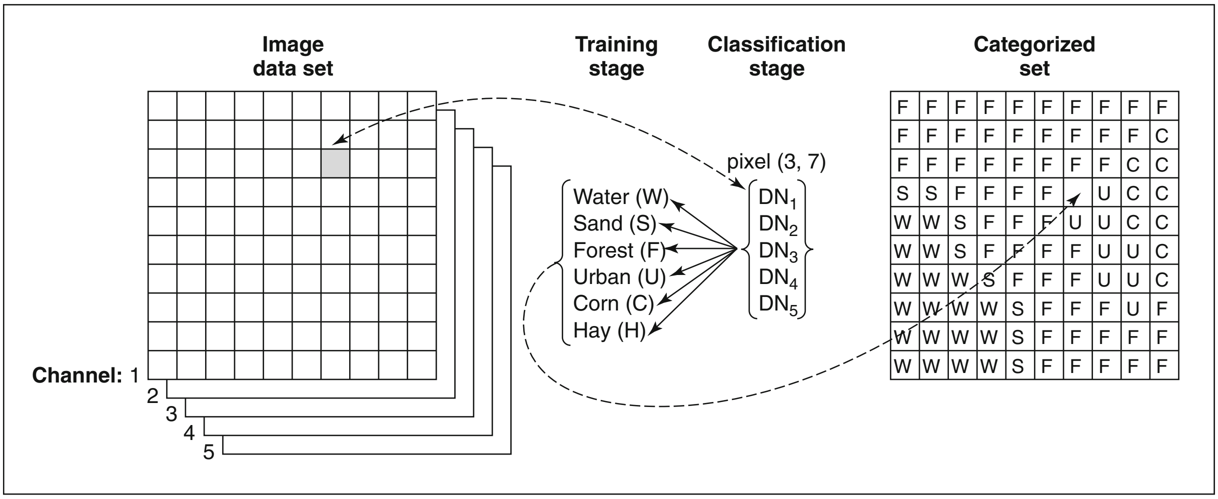
types. After the entire data set has been categorized, the results are presented in the output stage. Figure 6.22 illustrates the classification of an image into its land-cover types, including water, sand, forest, cornfield, and so forth and where the training stage fails, unclassified. Because of the presence of unclassified pixels, this methodology often requires a subjective allocation of these pixels into either their corn field neighbor or forest neighbor. To assist the analyst in making this subjective allocation, systematic procedures have been devised that will be described next.

B. Contextual Allocation of Pixels

The error in classification can come from different sources. It is reported, for example, that 50 percent of the light received by the scanner when pointing at one nominal pixel comes from nearby pixels (McLachlan 1992). Thus, much of the noise that corrupts the signal is spatially correlated. Since the whole observation process has a spatial component, there is a need to use contextual rules in allocating the pixels to the specified spatial groups. Contextual allocation rule means using a model that incorporates the a priori knowledge that spatially neighboring pixels tend to belong to the same group. With a contextual rule, a pixel is allocated not only on the basis of its observed feature vector, but also on the feature data of neighboring pixels. The use of a non-contextual rule that allocates a pixel j solely on the basis of its gray values, or its feature vector $\mathbf{x}_j$ representing its multispectral readings, and thereby ignores the information on neighboring pixels, leads to a patchwork quilt of colors representing the different disjoint groups (see Figure 6.24(b)). Oftentimes, contiguity of land use categories such as a lake or farmland, for example, is destroyed in the process.

One way of providing a contextual method of segmentation is to consider the allocation of each pixel individually on the basis of its posterior probabilities[12] of group membership given the recorded feature vectors x on all the n' pixels in the scene. Let $\tilde{\mathbf{z}}_j$ be the group-indicator vector defining the color of the jth-color

Figure 6.22 BASIC STEPS IN SUPERVISED CLASSIFICATION



SOURCE: Lillesand and Kiefer (1987). Reprinted with permission.

pixel with feature vector , where $\mathbf{z}_{ij} = 1$ if the jth-color pixel belongs to group i, G_i (i.e., $i = j$). Group i may represent a lake, farmland, and so on—the subregions of colors. The jth-color pixel is allocated then on the basis of the maximum of the posterior probability $P(\tilde{\mathbf{z}}_j = \mathbf{z}_j \,|\, \mathbf{x})$ with respect to $\mathbf{z}_j$, where $\mathbf{z}_j$ defines the group of origin color of the pixel. A common assumption is to form this posterior probability under the assumption of white noise, that is, the feature vectors $\mathbf{x}$ are conditionally independent given their group of origin color $\mathbf{z}_j$. Contextual rules that assume white noise offer less improvement in terms of error rate over non-contextual rules in situations where the feature data are spatially correlated. Overall, contextual rules still perform better than non-contextual rules even under this assumption.

We consider a binary example taken from Ripley (as reported by McLachlan [1992]). There are 2 groups representing two colors: white (G_1) and black (G_2). In the ith group G_i, each feature observation x_j is univariate normal with mean μ_i and variance σ^2 ($i = 1, 2$), where $\mu_1 = 0$ and $\mu_2 = 1$. An assumption on the prior distribution of the image is the Ising model, for which

$$P\,(z_{1j} = 1 \,|\, z_j) = \frac{\exp\,(\beta T_{1j})}{\exp\,(\beta T_{1j}) + \exp\,(\beta T_{2j})} \tag{6.6}$$

except at the edges, where T_{1j} is the number of white neighbors of the jth-color pixel, and $T_{2j} = 8 - T_{1j}$ is the number of black neighbors. In other words, the number of black and white neighbor pixels adds up to 8, considering both first-order and second-order neighbors. For known parameters μ_1, μ_2, σ^2, and β, we have from Bayes' Theorem[13] that

$$\log\left[\frac{P(z_{1j} = 1 \,|\, x,\, z_i)}{P(z_{2j} = 1 \,|\, x,\, z_i)}\right] = -\,\frac{\left(x_j - \frac{1}{2}\right)}{\sigma^2} + \beta\,(T_{1j} - T_{2j}) \tag{6.7}$$

Here the probabilities are conditioned upon a vector of feature pixel readings $\mathbf{x}$ from a sample band and the two-entry group indicator vector for the jth-color pixel, $\mathbf{z}_j$. An Iterative Conditional Mode (ICM) algorithm is devised whereby the jth-color pixel is allocated on the basis of Equation 6.7 where T_{ij} is replaced by its current estimate $\hat{T}_{1j}$ ($i = 1, 2$). Assuming equal posterior probabilities, the left hand side of Equation (6.7) is zero. Hence the jth-color pixel is allocated to white, or $z_{1j} = 1$, if

$$x_j < \frac{1}{2} + \beta\sigma^2\,(\hat{T}_{1j} - \hat{T}_{2j}) \tag{6.8}$$

which is both simple and intuitive. The non-contextual version of this rule, corresponding to $\beta = 0$, would take $z_{1j} = 1$ if $x_j < 1/2$, thereby ignoring the information $\hat{T}_{1j}$ and $\hat{T}_{2j}$ on the color of neighboring pixels. The algorithm can be extended to multivariate features (corresponding to multispectral bands) and multiple groups (colors) as will be illustrated below. (See McLachlan [1992] for further details beyond these two examples).

Two-Class Example

Consider a single-channel 42-pixel image with individual gray values as shown below (Brigantic and Chan 1994; Wright and Chan 1994):

11	5	6	8	13	2	8
2	5	1	1	3	2	3
8	4	6	5	6	6	6
1	4	2	2	6	3	3
4	5	6	5	5	4	5
3	2	2	1	5	2	3

We wish to classify each of these pixels as belonging to either a lake or forest. We will assume each class of pixels, whether lake or forest, are normally distributed with mean μ_i and standard deviation σ_i ($i = 1, 2$). Notice that instead of an overall, common standard deviation σ that applies to both groups, distinction is made between the two groups of pixels, σ_1 versus σ_2. The conditional probability-density function (PDF) of gray value x for the ith class ($i = 1, 2$) given the pixel is in the ith class $\mathbf{z}_i$ is therefore

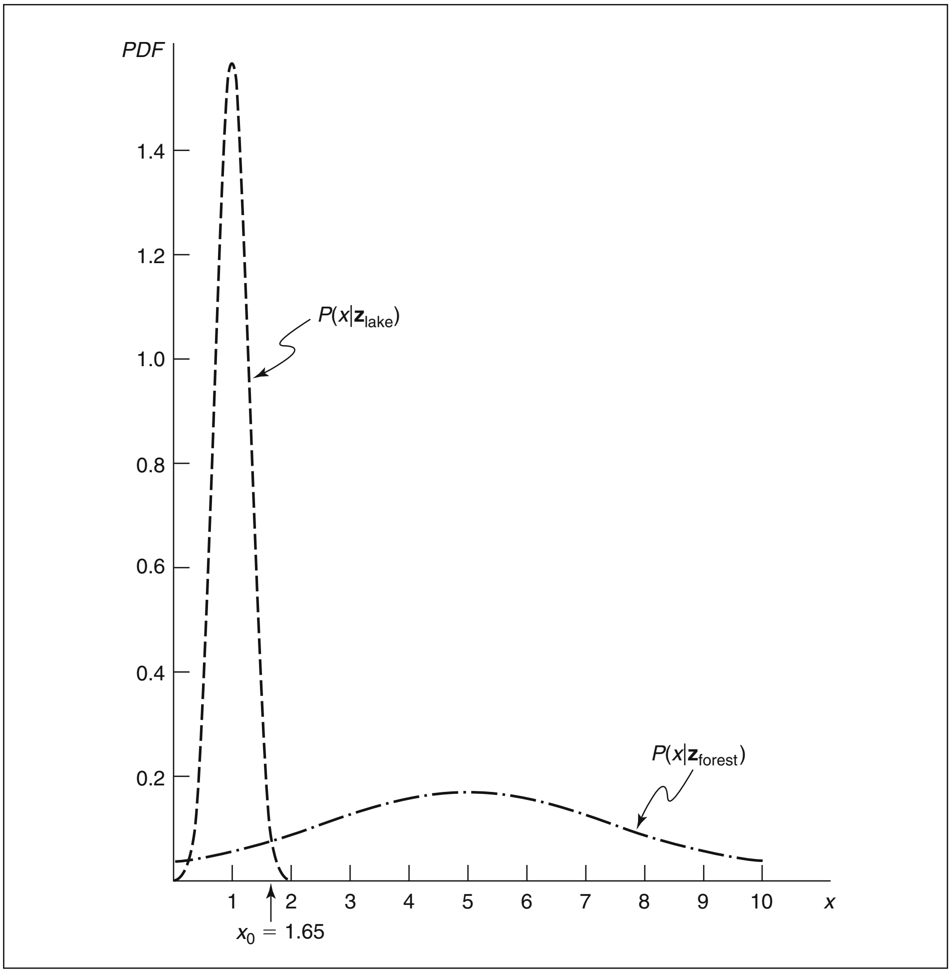
$$P(x|\mathbf{z}_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left[-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right] \qquad i = 1, 2 \tag{6.9}$$

Unless one class is more likely to occur, the point where the two PDFs are equal constitutes the decision boundary. We show an example of such PDFs in Figure 6.23, where we arbitrarily assume that the lake class has a mean gray value of $\mu_1 = 1$ and a standard deviation of $\sigma_1 = 0.25$. We also assume that the pixels with a gray value of other than 1 represent the forest class. Correspondingly, we compute a mean forest gray value of $\mu_2 = 4.74$ and a standard deviation of $\sigma_2 = 2.50$. From this plot we identify the decision boundary x_0 as 1.65. This means that any pixel with a gray value less than 1.65 is classified as lake and pixels with a gray value greater than 1.65 are classified as forest. The corresponding classified image is shown below, where F stands for forest and L for lake:

F	F	F	F	F	F	F
F	F	L	L	F	F	F
F	F	F	F	F	F	F
L	F	F	F	F	F	F
F	F	F	F	F	F	F
F	F	F	L	F	F	F

Notice this Bayesian classifier here simply computes the probability that a pixel belongs to a class, or it essentially performs a spectral classification (rather than a spatial classification). Contextual classification techniques are now applied to include the relation a pixel has to its neighbors. For example, if a lone pixel had a low gray value that indicates that it belongs to a lake, yet all of its first- and second-order neighbors had a high gray value suggestive of a forest, the contextual classification scheme may well assign the pixel to the

Figure 6.23 GAUSSIAN PROBABILITY-DENSITY FUNCTION USED IN BAYESIAN CLASSIFIER



forest class despite its low gray value. For our simple problem, we let first- and second-order neighbors have equal weights so that for interior pixels the number of first- and second-order neighbors total 8, or $T_{1j} + T_{2j} = 8$; for corner pixels $T_{1j} + T_{2j} = 3$; and for border pixels $T_{1j} + T_{2j} = 5$. According to Equation 6.8, the decision rule now assumes the form $x_j < 1.65 + \beta(0.25)^2(\hat{T}_{1j} - \hat{T}_{2j})$. So if pixel j had a gray value less than the resulting value as computed, it will be classified as a lake, otherwise it will be classified as forest. It is clear that the parameter β determines the watershed for classification. When $\beta = 0$, the gray values of a pixel's neighbors (and the choice of σ between the two groups) becomes unimportant, non-contextual classification results, as shown in the forest (F) and lake (L) image classification above.

Carrying out the ICM algorithm with values of β greater than zero, a number of classifications were obtained. At 0.25 increments, increasing the value from 0 through 1.5 did not cause a change in the classification. Starting at $\beta = 1.75$, however, the following image was obtained where two of the four lake pixels started to disappear:

F	F	F	F	F	F	F
F	F	F	F	F	F	F
F	F	F	F	F	F	F
L	F	F	F	F	F	F
F	F	F	F	F	F	F
F	F	F	L	F	F	F

Eventually, at $\beta = 2.25$ the predominance of forest pixels causes the lake pixels to disappear altogether:

F	F	F	F	F	F	F
F	F	F	F	F	F	F
F	F	F	F	F	F	F
F	F	F	F	F	F	F
F	F	F	F	F	F	F
F	F	F	F	F	F	F

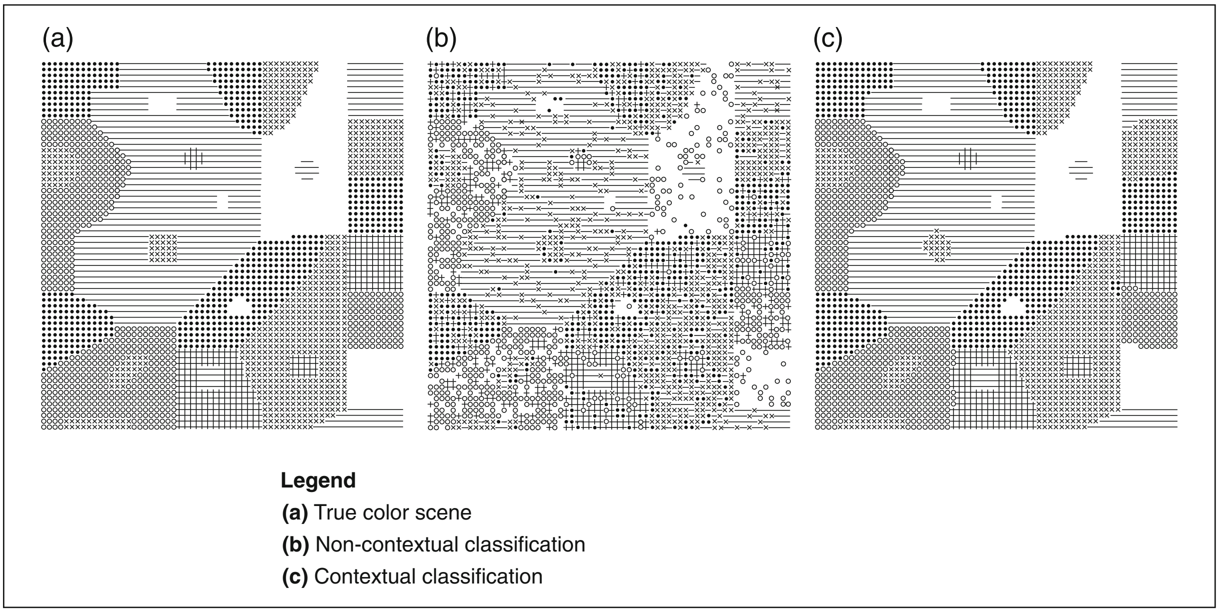
Notice that in terms of final result, the difference between a single σ vis-a-vis two σ's is really not that important since it gets to be combined with β in Equation 6.8.

Implementing the contextual classification scheme in conjunction with the Bayesian technique is relatively simple, at least for a single sensor (or one-dimensional problem) and in the case of partitioning pixels into two classes. Extension into several classes and multiple sensors is still straightforward, although computational requirements do go up noticeably, but not dramatically. ■

Multi-Class Example

In this example, we consider more than two classes in image classification. Take the example of four classes, the decision rule in Equation 6.8 is now based on three watershed points x_0, x_0', and x_0'', where $x_0 < x_0' < x_0''$. A pixel of color j will belong to group 1 if $x_j < x_0 + \beta\sigma_1^2(\hat{T}_{1j} - \hat{T}_{2j} - \hat{T}_{3j} - \hat{T}_{4j})$, to group 2 if $x_j < x_0' + \beta\sigma_2^2(\hat{T}_{2j} - \hat{T}_{1j} - \hat{T}_{3j} - \hat{T}_{4j})$, and to group 3 if $x_j < x_0'' + \beta\sigma_3^2(\hat{T}_{3j} - \hat{T}_{1j} - \hat{T}_{2j} - \hat{T}_{4j})$. While there are numerous ways to classify an image, one way is that the classified pixels are sequentially removed from the image according to this decision rule. Thus after pixels are classified into group 1, they are removed and $\hat{T}_{1j} = 0$ in the decision rule for classifying group 2 pixels. If one does not want to remove pixels from an image after they are classified, an alternate decision rule can be devised. The rule for group 1 remains the same, while that for group 2 becomes $x_j < x_0 + \beta\sigma_2^2(\hat{T}_{1j} + \hat{T}_{2j} - \hat{T}_{3j} - \hat{T}_{4j})$ and that for group 3 becomes $x_j < x_0'' + \beta\sigma_3^2(\hat{T}_{1j} + \hat{T}_{2j} + \hat{T}_{3j} - \hat{T}_{4j})$. The decision rules can easily be generalized to six groups in the following example. We give an artificial example taken from Besag as cited in McLachlan (1992). The true scene contains 6 colors, on a 120 × 120 array. It was originally hand-drawn and chosen to display a wide variety of characteristics. The univariate feature observations were generated from the color labels by superimposing Gaussian noise with $\sigma^2 = 0.36$. The first 64 rows and the first 64 columns are displayed in Figure 6.24(a) in which the adjacencies are less contrived than in the scene as a

Figure 6.24 CONTEXTUAL VERSUS NON-CONTEXTUAL IMAGE CLASSIFICATION


(a)
(b)
(c)
Legend
(a) True color scene
(b) Non-contextual classification
(c) Contextual classification

SOURCE: Besag as cited in McLachlan (1992). Reprinted with permission.

whole. A color key, which is part of the pattern, is shown, where the sign "minus" − = 1, "cross" × = 2, and so on. The initial reconstruction using the non-contextual classification, i.e., $\beta = 0$, produced an overall misallocation rate of 32 percent, as shown in the patchwork of colors in Figure 6.24(b). With the correct value of σ^2 and with $\beta = 1.5$ throughout, the ICM algorithm gave an overall error rate of 2.1 percent on the eighth cycle. The ICM algorithm, applied with β increased by equal increments from 0.5 to 1.5 over the first six cycles, reduced the overall error rate to 1.2 percent on the eighth cycle, as shown in Figure 6.24(c). ■

IX. A DISTRICT CLUSTERING MODEL

Following the same philosophy, we present here a more general spatial pattern recognition model—sometimes called a districting model—which selects, from a set of candidate pixels, parcels, or cells, the collection that best achieves a pre-specified set of goals or objectives. This is accomplished within specific constraints. The main purpose of most districting models is the design of a predetermined number of territories or districts with contiguous and compact shapes. This compares well with image classification, where we group like pixels together to identify land cover or manmade facilities of interest.

A. A Single Subregion Model

Benabdallah and Wright (1992) present a multi-criteria integer-programming model for selecting a set of cells or parcels from a large set, and identify the complete set of non-inferior or non-dominated[14] solutions on a regular grid configuration. A binary variable x_i takes on a value of 1 if cell i is acquired and 0 otherwise. If we assume that the cost c_i for any particular cell being considered is known, then a criterion function that seeks to minimize overall cost may be written: Minimize $y_2' = \Sigma_{i=1}^{n'} c_i x_j$. Here costs are broadly defined to include such measures as the gray value of a pixel or subareal population exposed to pollution and so forth. For the example shown in Figure 6.23, application of this criterion function will most likely result in the identification of the previously unclassified pixels as forest. A maximization criterion, on the other hand, will probably identify the pixels as corn.

Similarly, if a_i'—the area of cell i—is also known for all candidate cells, a second criterion function that seeks to maximize total area may be written; Maximize $y_2' = \Sigma_{i=1}^{n'} a_i' x_i$. The application of this criterion function will support our conjecture that the unclassified pixels in Figure 6.23 will either be identified as forest or corn and not be split between the two, since this criterion fosters as large an acquisition as possible for a land cover type.

A third criterion function may be to induce compactness of the area by tightening the total length of the border surrounding the cells acquired. This premise was based on the observation that, for any given area, the most compact configuration possible is one in which the border surrounding that area is a circle, the border with the shortest length. Define s_{ij} as the length of the border separating cell i from cell j, and variables P_{ij} and N_{ij} to be mutually exclusive binary decision variables that sum to 1 if the border separating cells i and j in the final solution is an external border (a border separating a cell that is acquired from one that is not) and 0 otherwise. The compactness criterion may be written as: Minimize $y_3' = \Sigma_{i=1}^{n'} \Sigma_{j \in T_i}^{n'} s_{ij}(P_{ij} + N_{ij})$, where T_i is the set of cells adjacent to cell i, in other words, if cell 11 is adjacent to first-order neighbor cells 7, 10, 12, and 15, then the set T_{11} is {7, 10, 12, 15}.

The combined weighted objective function for the three criterion districting problem is

$$\text{Minimize } z = \lambda_c' \sum_{i=1}^{n'} c_i x_i - \lambda_a' \sum_{i=1}^{n'} a_i x_i + \lambda_s' \sum_{i=1}^{n'} \sum_{j \in T_i} s_{ij} (P_{ij} N_{ij}) \tag{6.10}$$

where λ_c', λ_a', λ_s' are weights on the cost, area, and compactness objectives, respectively. By varying the weights, the three criterion functions are emphasized or de-emphasized relative to one another. Thus emphasizing the gray value of the unclassified pixels in Figure 6.23, balanced with maximal acquisition, may result in only part (rather than all) of the pixels being allocated to forest.

A single set of constraints is required to define P_{ij} and N_{ij}

$$x_i - x_j - P_{ij} + N_{ij} = 0 \qquad \forall\, i, j \in T_i \tag{6.11}$$

For any adjacent cells i and j, if only one of the cells is selected ($x_i = 1$, $x_j = 0$; or $x_i = 0$, $x_j = 1$), then either P_{ij} and N_{ij} must equal 0. For example, if cell i is acquired ($x_i = 1$) and cell j is not ($x_j = 0$), then the above equation would be satisfied if $P_{ij} = 1$ and $N_{ij} = 0$. If both cells i and j are acquired ($x_i = x_j = 1$) or neither is acquired ($x_i = x_j = 0$), then P_{ij} and N_{ij} must both equal 1 or both equal 0. Because the external border function is being minimized, the smallest values assigned to P_{ij} and N_{ij} that would satisfy the equation would be 0 ($P_{ij} = N_{ij} = 0$) for all i, j within the same grouping. (Example: If only cell 11 is acquired, or $x_{11} = 1$, such a non-inferior solution will have $P_{11\ 7} = P_{11\ 10} = P_{11\ 12} = P_{11\ 15} = 1$, and $N_{7\ 11} = N_{10\ 11} = N_{12\ 11} = N_{15\ 11} = 1$.)

While the formulation presented above is a general model, it is computationally difficult due to integrality requirements on the decision variables. Including the second and third criteria as constraints[15], a more compact formulation is obtained:

$$\text{Minimize } y_1' = \sum_{i=1}^{n'} c_i x_i \tag{6.12}$$

subject to

$$\sum_{i=1}^{n'} x_i = M \tag{6.13}$$

$$x_i - x_j - P_{ij} + N_{ij} = 0 \qquad \forall\, i, j \in T_i \tag{6.14}$$

$$\sum_{i=1}^{n'} \sum_{j \in T_i} s_{ij} (P_{ij} + N_{ij}) = L \tag{6.15}$$

$$x_i,\, P_{ij},\, N_{ij} \in \{0, 1\} \tag{6.16}$$

Here, M stands for the number of pixels to be included in the district, and L is twice the boundary of the district[16]—both of which are to be parametrically varied within a range to reflect the change in weights λ_a' and λ_s'. All a_i's are set to unity since pixels are equal in size. The advantage of this formulation is that a noninferior solution set (or efficient frontier) can be traced out for the allowable range of Equations 6.13 and 6.15. (Again, see Section III in Chapter 5 for details of such a procedure.)

We will illustrate this transformed model through a numerical example. Figure 6.25 shows a 4 × 4 grid of 16 unit squares of the same size. The costs are shown as lower right-hand-side entries of the grid. The cell (or pixel) number is at the left-hand upper corner of each cell. In this example, $M = 2$, indicating an area of 2 units. Also, $L = 12$ represents a boundary line of 6 units in length. The entire formulation is shown in Figure 6.26, prepared in an ASCII input format

Figure 6.25 A NONINFERIOR SOLUTION SHOWING A SINGLE SUBREGION

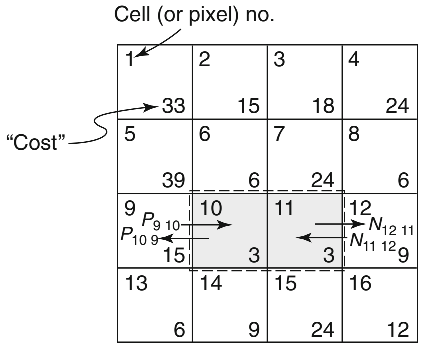


Figure 6.26 EXAMPLE MODEL FORMATION

```
. . OBJECTIVE MINIMIZE
1 ( 33 [ [ x1] ] + 15 [ [ x2] ] + 18 [ [ x3] ] + 24 [ [ x4] ] +
39 [ [ x5] ] + 6 [ [ x6] ] + 24 [ [ x7] ] + 6 [ [ x8] ] +
15 [ [ x9] ] + 3 [ [ x10] ] + 3 [ [ x11] ] + 9 [ [ x12] ] +
6 [ [ x13] ] + 9 [ [ x14] ] + 24 [ [ x15] ] + 12 [ [ x16] ] )
CONSTRAINTS
*constraint for sum (Xi) = M
x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11 + x12 +
x13 + x14 + x15 + x16 = 2

*constraint for Xi − Xj − Pij + Nij = 0

x1 − x2 − p12 + n12 = 0
x1 − x5 − p15 + n15 = 0
x2 − x1 − p21 + n21 = 0
x2 − x3 − p23 + n23 = 0
x2 − x6 − p26 + n26 = 0
x3 − x2 − p32 + n32 = 0
x3 − x4 − p34 + n34 = 0
x3 − x7 − p37 + n37 = 0
x4 − x3 − p43 + n43 = 0
x4 − x8 − p48 + n48 = 0
x5 − x1 − p51 + n51 = 0
x5 − x6 − p56 + n56 = 0
x5 − x9 − p59 + n59 = 0
x6 − x2 − p62 + n62 = 0
x6 − x5 − p65 + n65 = 0
x6 − x7 − p67 + n67 = 0
x6 − x10 − p610 + n610 = 0
x7 − x3 − p73 + n73 = 0
x7 − x6 − p76 + n76 = 0
x7 − x8 − p78 + n78 = 0
x7 − x11 − p711 + n711 = 0
x8 − x4 − p84 + n84 = 0
x8 − x7 − p87 + n87 = 0
x8 − x12 − p812 + n812 = 0
x9 − x5 − p95 + n95 = 0
x9 − x10 − p910 + n910 = 0
x9 − x13 − p913 + n913 = 0
x10 − x6 − p106 + n106 = 0
x10 − x9 − p109 + n109 = 0
x10 − x11 − p1011 + n1011 = 0
x10 − x14 − p1014 + n1014 = 0
x11 − x7 − p117 + n117 = 0
x11 − x10 − p1110 + n1110 = 0
x11 − x12 − p1112 + n1112 = 0
x11 − x15 − p1115 + n1115 = 0
x12 − x8 − p128 + n128 = 0
x12 − x11 − p1211 + n1211 = 0
x12 − x16 − p1216 + n1216 = 0
x13 − x9 − p139 + n139 = 0
x13 − x14 − p1314 + n1314 = 0
x14 − x10 − p1410 + n1410 = 0
x14 − x13 − p1413 + n1413 = 0
x14 − x15 − p1415 + n1415 = 0
x15 − x11 − p1511 + n1511 = 0
x15 − x14 − p1514 + n1514 = 0
x15 − x16 − p1516 + n1516 = 0
x16 − x12 − p1612 + n1612 = 0
x16 − x15 − p1615 + n1615 = 0
*constraint for sum (Pi + Ni)  = L
p12 + n12 + p15 + n15 +
p21 + n21 + p23 + n23 + p26  + n26 +
p32 + n32 + p34 + n34 + p37  + n37 +
p43 + n43 + p48 + n48 +
p51 + n51 + p56 + n56 + p59  + n59 +
p62 + n62 + p65 + n65 + p67  + n67 + p610  + n610 +
p73 + n73 + p76 + n76 + p78  + n78 + p711  + n711 +
p84 + n84 + p87 + n87 + p812  + n812 +
p95 + n95 + p910 + n910 + p913  + n913 +
p106 + n106 + p109 + n109 + p1011  + n1011 + p1014  + n1014 +
p117 + n117 + p1110 + n1110 + p1112  + n1112 + p1115  + n1115 +
p128 + n128 + p1211 + n1211 + p1216  + n1216 +
p139 + n139 + p1314 + n1314 +
p1410 + n1410 + p1413 + n1413 + p1415  + n1415 +
p1511 + n1511 + p1514 + n1514 + p1516  + n1516 +
p1612 + n1612 + p1615 + n1615 = 12
```

that will be accepted by many mixed integer programming codes on the market today. The solution, $x_{10} = x_{11} = 1$ and the rest of the variables at zero value, illustrates only one non-inferior solution for the criterion function Equation 6.12 when M and L in Equations 6.13 and 6.15 assume specific values of 2 and 6 respectively. In the general case, there are quite a few number of partitioning possible within permissible range of M and L.

To determine the range of L and M to vary in this constrained feasible region, Benabdallah and Wright (1992) offered a formula and subsequently modified by Wright (1994) for determining the minimum value of the L range as a function of M: $B^M_{\text{Min}} = 4\langle\sqrt{M}\rangle + 2t''$ with

$$t''(M) = \begin{cases} 0 \textit{ if } M - \langle\sqrt{M}\rangle^2 = 0 \\ 1 \textit{ if } M \leq \langle\sqrt{M}\rangle^2 + \langle\sqrt{M}\rangle \\ 2 \textit{ if } M > \langle\sqrt{M}\rangle^2 + \langle\sqrt{M}\rangle \end{cases} \tag{6.17}$$

where <•> is the integer part of • The parameter $t''(M)$ monitors the shape of the region, when $t''(M) = 0$, the region is square in shape. When $t''(M) = 1$, the shape becomes a rectangle. For example, an irregular shape will result for $M = 3$, $t'' = 2$, and $B^M_{\text{Min}} = 8$: ■■ □■. Similar result is obtained for $M = 8$, $t''(M) = 2$, and $B^M_{\text{Min}} = 12$: ■■■ ■■■ □■■. When $t''(M) = 2$, the shape will be made up of squares and/or rectangles. Experimentation with small problems will show that the above equations make sense and that for values of L near this lower bound, contiguity of the region will result. When the strict lower bound is used for L, a rectangular or square shape subregion will be formed.

Example
Refer to the numerical example of Figure 6.26 and Figure 6.27. For $M = 2$, and $t'''(M) = B^M_{\text{Max}} = 6$, and the familiar rectangle consisting of cells 10 and 11 results. The maximum of the range is $B^M_{\text{Max}} = 4M$, where the region is fragmented and no acquired cell is adjacent to any other acquired cell □■□ ■□■ □■□. In general, the region may be fragmented before reaching $B^M_{\text{Max}} = 4M$. ■

B. Multiple Subregion Model

A multiple subregion model can be obtained by introducing another index k, which stands for the subregion number. Thus x_{ik} stands for the binary variable that determines whether the ith cell or pixel is acquired in subregion k. Correspondingly, the mutually exclusive binary-variables associated with the boundary can be extended to include the index k: P_{ijk} and N_{ijk}. Introducing the weight w_k for each subregion, the model now looks like

$$\text{Minimize} \quad z = \sum_{k=1}^{K} w_k \sum_{i=1}^{n'} c_i \, x_{ik} \tag{6.18}$$

Figure 6.27 MULTIPLE SUBREGION NONINFERIOR SOLUTIONS

File[1]	M_1	L_1	Cost	Cells	M_2	L_2	Cost	Cells	Total Cost
S2_1_1	1	4	1	8	1	4	1	9	2
S2_1_2	1	4	1	8	2	6	3	6, 9	4
S2_1_3	1	4	5	2	3	8	4	6, 8, 9	9
S2_1_4a	1	4	5	2	4	8	17	5, 6, 8, 9	22
S2_1_4b	1	4	5	2	4	10	9	6, 7, 8, 9	14
S2_1_5a	1	4	5	7	5	10	22	2, 5, 6, 8, 9	27
S2_1_5b	1	4	5	7	5	12	15	2, 3, 6, 8, 9	20
S2_1_6a	1	4	5	7	6	10	28	2, 3, 5, 6, 8, 9	33
S2_1_6b	1	4	6	3	6	12	27	2, 5, 6, 7, 8, 9	33
S2_1_6c	1	4	5	2	6	14	23	3, 4, 6, 7, 8, 9	28
S2_1_7a	1	4	6	3	7	12	35	2, 4, 5, 6, 7, 8, 9	41
S2_1_7b	1	4	2	6	7	14	39	2, 3, 4, 5, 7, 8, 9	41
S2_1_7c	1	4	1	8	7	16	38	1, 2, 3, 4, 6, 7, 9	39
S2_1_8a	1	4	1	9	8	12	51	1, 2, 3, 4, 5, 6, 7, 8	52
S2_1_8b	1	4	2	6	8	14	50	1, 2, 3, 4, 5, 7, 8, 9	52
S2_1_8c	1	4	13	5	8	16	39	1, 2, 3, 4, 6, 7, 8, 9	52
S2_2_2	2	6	3	6, 9	2	6	6	7, 8	9
S2_2_3	2	6	6	7, 8	3	8	9	3, 6, 9	15
S2_2_4a	2	6	11	2, 3	4	8	17	5, 6, 8, 9	28
S2_2_4b	2	6	11	2, 3	4	10	9	6, 7, 8, 9	20
S2_2_5a	2	6	11	2, 3	5	10	22	5, 6, 7, 8, 9	33
S2_2_5b	2	6	13	4, 7	5	12	15	2, 3, 6, 8, 9	28
S2_2_6a	2	6	13	4, 7	6	10	28	2, 3, 5, 6, 8, 9	41
S2_2_6b	2	6	2	8, 9	6	12	39	2, 3, 4, 5, 6, 7	41
S2_2_6c	2	6	2	8, 9	6	14	37	1, 2, 3, 4, 6, 7	39
S2_2_7a	2	6	19	1, 4	7	12	33	2, 3, 5, 6, 7, 8, 9	52
S2_3_3	3	8	7	7, 8, 9	3	8	13	2, 3, 6	20
S2_3_4a	3	8	7	7, 8, 9	4	8	26	2, 3, 5, 6	33
S2_3_4b	3	8	13	2, 3, 6	4	10	15	4, 7, 8, 9	28
S2_3_5a	3	8	7	7, 8, 9	5	10	34	2, 3, 4, 5, 6	41
S2_3_5b	3	8	9	3, 6, 9	5	12	30	1, 2, 4, 7, 8	39
S2_3_6a	3	8	22	1, 2, 3	6	10	30	4, 5, 6, 7, 8, 9	52
S2_3_6b	3	8	4	6, 8, 9	6	12	48	1, 2, 3, 4, 5, 7	52
S2_3_6c	3	8	26	4, 5, 7	6	14	26	1, 2, 3, 6, 8, 9	52
S2_4a_4b	4	8	26	2, 3, 5, 6	4	10	15	4, 7, 8, 9	41
S2_4a_5b	4	8	17	5, 6, 7, 8	5	12	35	1, 2, 3, 4, 7	52

[1] Take the first entry under this column, S2 stands for 2 subregions, 1 stands for an area of 1 pixel for subregion 1 and the last 1, stands for an area of 1 pixel for subregion 2 also. The a and b entries specify two different variations on the boundary of the subregion in generating noninferior solutions.

subject to

$$\sum_{i=1}^{n'} x_{ik} = M_k \quad \forall k \tag{6.19}$$

$$x_{ik} - x_{jk} - P_{ijk} + N_{ijk} = 0 \qquad \forall\, i, j \in T_i, k \tag{6.20}$$

$$\sum_{k=1}^{K} x_{ik} \le 1 \qquad \forall k \tag{6.21}$$

$$\sum_{i=1}^{n'} \sum_{j \in T_i} s_{i_j} \, (P_{ijk} + N_{ijk}) = L_k \qquad \forall k \tag{6.22}$$

The number of zero-one variables and constraints used in the multiple subregion models can be estimated a priori before one runs the model. Let $\overline{R}$ be the number of cell rows in the overall region, $\overline{C}$ the number of cell columns in the overall region, and K the number of subregions being acquired. An estimate of the number of zero-one variables, the number of equality constraints, and less-than-equal-to constraints as a function of $\overline{R}$, $\overline{C}$, and K can be given.

Number of zero-one variables $\approx 9K\overline{R}\overline{C}$
Number of equality constraints $\approx 2K(1 + \overline{R} + \overline{C} + 2\,\overline{R}\,\overline{C}\,)$
Number of less-than-or-equal-to constraints $\approx \overline{R}\,\overline{C}$
An example calculation when $\overline{R} = 3$, $\overline{C} = 3$, and $K = 2$ yields 162, 100, and 90 respectively. It can be seen that the size of the problem can grow exponentially large. Either a faster solution algorithm or an alternate model formulation would be necessary to make this an operational procedure.

An example run involving two subregions is shown in Figure 6.27. This table is organized into four groups, corresponding to the size of the first subregion fixed at 1, 2, 3, and 4 pixels respectively, while varying the size of the second subregion. Also included in this table is the various non-inferior solutions when the boundary L is tightened or loosened. This illustrates the usefulness of a model like this in presenting the analyst with various possible cell classifications schemes. The decision maker can then pick and choose among the non-inferior solutions. Figure 6.28 illustrates graphically the non-inferior solutions for the first group in the table. Take the line marked S2_1_2 as an example. The line records a single non-inferior solution to a *two*-subregion model. The first subregion has *one* pixel while the second subregion has *two* pixels. The partitioning is based on the assumption that the first subregion is weighted twice more than the second subregion. Instead of allocating pixel 6 to subregion 1 and pixels 8 and 9 to subregion 2, it can be shown that the better solution is to have pixel 8 assigned to subregion 1 and pixels 6 and 9 to subregion 2. To see this, let us say $w_1 = 1$ and $w_2 = 2$, the solution as shown yields an objective function of 2(1) + (2 + 1) = 5, which is better than 2(2) + (1 + 1) = 6.

The model can be further extended to account for shape of a subregion. Let w'_k be the width and h_k be the height of a subregion k. One can specify the shape of each subregion by rewriting Equation 6.22 as two equations:

$$\sum_{i \in n^r} x_{ik} - W'_k y_{rk} = 0 \qquad \forall\, r, k \tag{6.23}$$

$$\sum_{r=1}^{\overline{R}} y_{rk} = h_k \qquad \forall k \tag{6.24}$$

where y_{rk} is a binary variable equal to 1 if any cell in row r is assigned to subregion k and zero otherwise. The parameter n_r is the set of cells in row r and $\overline{R}$ is the number of rows in the grid. An example of this model is illustrated in

Figure 6.28 MULTIPLE SUBREGION ALLOCATION RESULTS

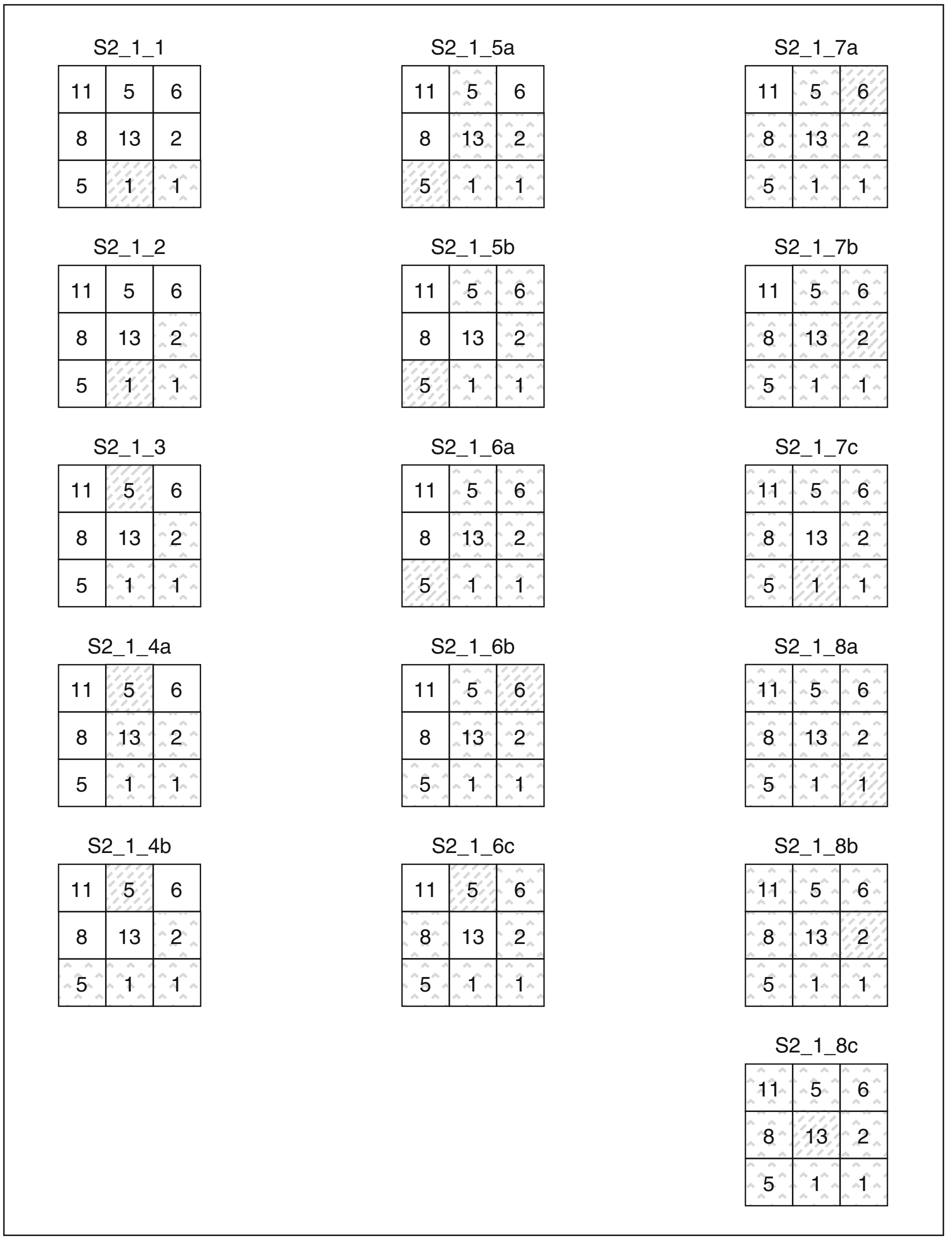
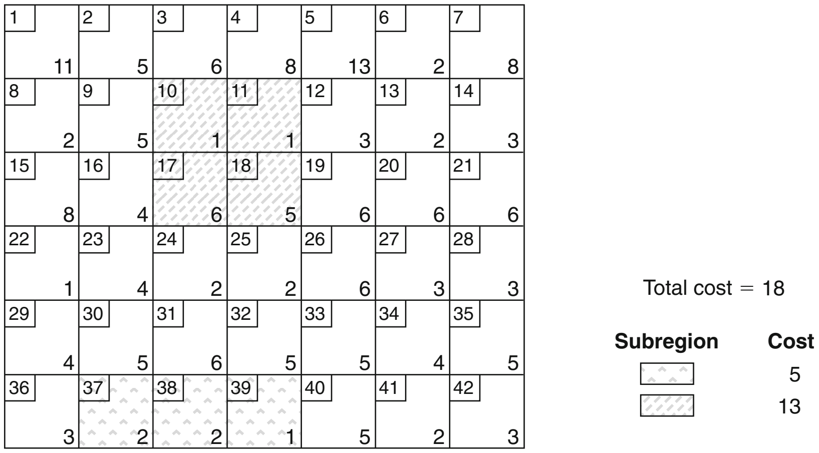


Figure 6.29, in which the first subregion is specified to have a width of 2 and a height of 2, while the second subregion measures 3 by 1. Notice this fundamental formulation is good for subregions of rectangular and square shapes only, where the solution yields the exact specified shape.

For computational efficiency, this model has been transformed to several more compact formulations. For example, a nonlinear function can be used as a compactness function for a districting model (Benabdallah and Wright 1992). The resulting model is multi-criteria, nonlinear, and discrete. The objective of

Figure 6.29 MULTIPLE SUBREGION MODEL WITH SHAPE SPECIFICATIONS

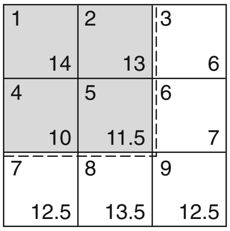


the model is to maximize the weighted sum of the compactness function of all subregions, subject to the cost limits constraint on each subregion. A heuristic algorithm is developed to generate a solution to the problem. Based on limited experiments, the algorithm converges to a very good solution. However, the solution may not be optimal.

C. Demand Equity

In providing service to a region, the concept of equity is important. This is particularly true in districting for public service provision. The concept of equity asserts that the entire population of potential clients is treated as equally as possible in terms of the quality of service it receives. Applying the equity criterion will imply that the performance measures by which the quality of service is evaluated will be more or less equal in each subregion. Let us examine the sample region G exhibited in Figure 6.30. Instead of c_i, the region consists of nine cells. The numbers at the lower right-hand corner of each cell indicate the demand at each cell. These are denoted by f_i ($i = 1, \ldots, 9$). Suppose we want to partition G into two subregions, G_1 and G_2, where the only guiding criterion is equity. We will certainly not recommend

Figure 6.30 DISTRICTING FOR DEMAND EQUITY



SOURCE: Ahituv and Berman (1988). Reprinted with permission.

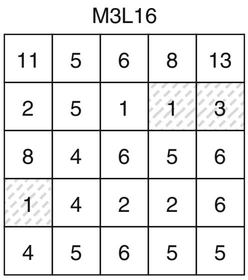
that cells 2 and 3 constitute G_1 while all the rest of the cells are assigned to G_2, since such partitioning will load 81 percent of the total demand on G_2. Rather we will try to divide the cells such that their cumulated demand will be close to 50 percent. For example, $G_1 = \{1, 2, 4, 5\}$ and $G_2 = \{3, 6, 7, 8, 9\}$; this partitioning will split the demand between the two subregions in a ratio of 48.5:51.5.

The principle of equity can be quantitatively formulated as follows: Let K be the desired number of subregions. Perfect equity is obtained if each subregion incurs $1/K$ fraction of the total demand. An additional criterion function to be applied, in addition to compactness, cost, and area, may well be the widely publicized entropy function[17] and Max $Q!/\Pi_{k=1}^{k} V_k$ where $V_k = \Sigma_{i=1}^{n'} f_i x_i$, $\Sigma_{i=1}^{k} V_k = Q$ and f_i's are integer valued. Obviously, such a criterion function is nonlinear, even when it is simplified into its Stirling's approximation: Max $\left[-\sum_{i=1}^{k} (V_k \log V_k - V_k)\right]$. The set of constraints are very much similar to Equations 6.19 through 6.22. The model will then partition the study area into service regions of more or less equal demand. Unfortunately, the resulting model is nonlinear, discrete, and huge in size. For this reason, it may further complicate the already computationally demanding Benabdallah/Wright (BW) model. Simpler districting models without the area and border length considerations have been around. They are typically used for effecting equitable redistricting of political subdivisions. Mehrotra and Johnson (1995) provides one of the more recent descriptions of a solution algorithm. A numerical example is included in the "Exercise and Problems" section as Problem III-A.

D. Extensions

The BW model can be further extended in several ways. First, there is an inherent weakness in handling subregions at the border of the grid. The accounting system of the model breaks down at the border. For instance, the border length of a subregion made up of cells 13 and 14 in Figure 6.25 is 3 rather than 6, since the edges at the border do not count. Also a subregion can be broken down into two at the border. The example shown in Figure 6.31 illustrates this fact, where the shaded cells 9, 10, and 16 form one (rather than two) subregion(s) of area 3 and border length 8 ($M = 3$, $L = 16$). This weakness of the model can be overcome by rewriting the equations governing the subregion length L, distinguishing between the regular interior cells, the corner cells and cells on the border that are not corner cells. A simpler way is to build an artificial border around the region, with values c_i set at a high value. This way, each real cell can be treated the same way without

Figure 6.31 A SPLIT SUBREGION AT THE BORDER



M3L16

11	5	6	8	13
2	5	1	1	3
8	4	6	5	6
1	4	2	2	6
4	5	6	5	5

having to distinguish between interior cells, border cells and corner cells. This practice also parallels remote sensing applications, where the pixels at the border are distorted and are of little relevance to the rest of the image.

If there is a single criticism leveled against the BW model, it is about the computational time involved. The current state of the art only allows such a model to be a research tool, rather than an operational one. It is conceivable that better solution algorithms can be devised over time to address this problem. Finally, the shape constraint can possibly be made more elaborate by inclusion of more sophisticated constraints. Research is currently underway to address some of these concerns (Green and Chan 1994; Warrender et al. 1992). Modern GIS technology has developed to the point where exhaustive enumeration algorithms imposed on raster or cell data can solve problems that are much more practical, and in reasonable time, even though these exhaustive algorithms are by definition inefficient. However, for problems that involve clustering of all cells in a field into distinct subregions and the identification of multiple subregions having certain shape or configuration requirements enumeration methods are infeasible.

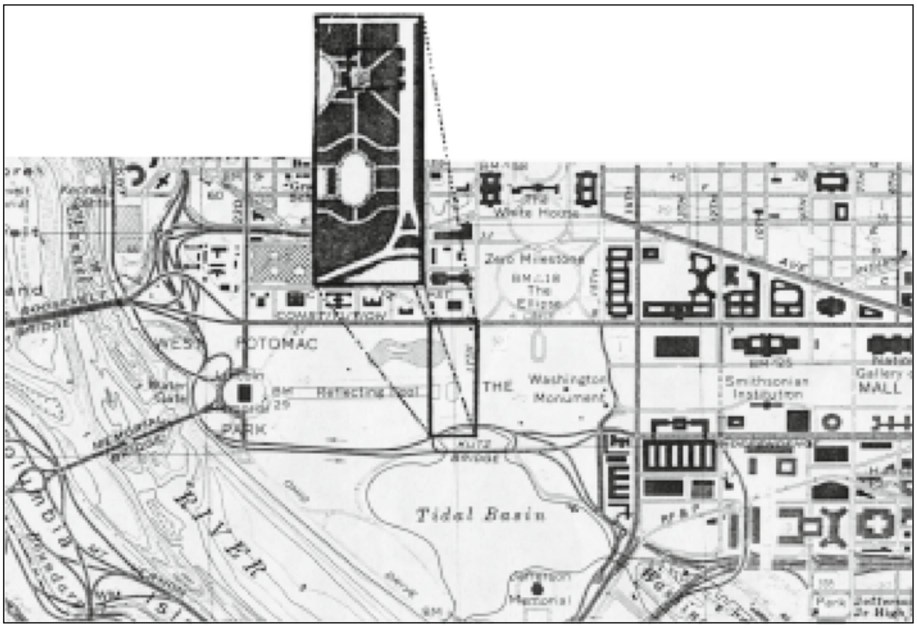
X. CASE STUDY OF IMAGE CLASSIFICATION

For this study, a SPOT image of the Washington D.C. area was used as the source of multi spectral data. Land cover types are to be discerned in the image. Rather than attempting to analyze the whole image, this study will be limited to a 48 × 18 pixel sub-image. The area selected is a portion of the Washington D.C. Mall located between the Lincoln Memorial and the Washington Monument (see Figure 6.32). Our objective is to classify the bodies of water found in the Reflecting Pool, Tidal Basin, and Constitution Gardens. The area is chosen because the ground truth regarding the bodies of water is well-known from maps, serving to validate any of our classifications. All three multispectral channels of this sub-image will be used in the analysis. In each channel, the individual pixels are allowed to take on one of 256 shades of gray (Amrine 1992).

A. Digital Image Data

Two assumptions were made regarding this image: (a) No rectification was needed among each of the multispectral images. (b) The processing effects on the image are minimal and do not affect the analysis. The software TS-IP[18] unveils several interesting observations in the images of Figure 6.33 regarding the spectral values. In channel 1, the values ranged from 0 to 227, while in channel 2, the range was 0 to 216, and in channel 3, from 0 to 212. From the processed sub-image based on spectral filters, it is obvious that the spectral range used by these filters identifies water, but not uniquely. For example, the filter uses a spectral range of 0–22 to identify water in channel 1. Table 6.6 documents the overall accuracy in identifying the water contained in the four water subregions of the channel-1 image as 94 percent. In channel 2, the water subregions are not as spectrally distinct. The visual inspection method used in channel 1 could not be used for channel 2. It is first necessary to locate the water subregions from the ground truth so the gray values could be recorded. It is found that the spectral range of water is 5–166. This wide range for

Figure 6.32 PORTION OF WASHINGTON D.C. MALL UNDER ANALYSIS



Note: ┌ ┐ represents an area for further analysis in the "Spatial-Temporal Information" chapter in Chan (2005).

SOURCE: U.S. Geological Survey (1983). Reprinted with permission.

Figure 6.33 SPOT SUB-IMAGE GRAY VALUES

Channel 1

```
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 224 222 211 213 213 219 221 217 198 211 204 157 174 145 139 168  90  58 999
999 168 192 204 210 213 210 198 214 211 204 211 174 186 192 163 163  82 107 999
999 121 139 139 139 174 157 133 204 168 121 174 163 107 127 168 145  82 157 999
999 192 168 163 192 198 174 174 211 174 121 186 127  90 145 192 121  74 180 999
999 157 145 139 151 163 139 174 210 168 139 157 139 186 204 163 107  82 139 999
999  25  99 133  99 133 157 180 204 180 168 157 211 214 151  99  99  74  90 999
999   0  33 145 151 151 157 151 213 151 151 192 204 192  74 145 157  58  90 999
999   0   0  74 133 115 133 174 210 115 145 192 174 139 121 192 157  99 145 999
999   0   0  16 133 139 151 180 192 192 192 168 145 163 186 180 145  82 145 999
999   0   0   0 139 168 157 186 192 215 168 115 151 192 180 139 115 107 157 999
999   0   0   0 121 145 163 180 192 210 115 151 174 210 180 133 115  82  99 999
999   0   0   6 121 157 168 192 210 218 174 145 192 204 157 168 107 107 121 999
999   0   0  22 157 157 174 213 215 216 211 163 127 127 127 168 133 127 163 999
999   0   0 133 168 115  99 174 174 133 192 192 139  82 127 174 127 139 180 999
999   0 107 180 139 139 139 198 204 127 163 210 198 151 115 163 107 115 151 999
999 180 198 107 107 145 174 186 214 180 151 186 213 215 180 157  90  74 121 999
999 214 107  58 151 151 145 168 211 163 157 127 145 211 198 168  99  74 139 999
999 157 115 127 151 139 145 186 192 163 145 145 127 157 204 180 107 107 168 999
999 139 151 133 157 180 180 163 192 210 180 163 168 163 168 192 127 145 163 999
999 204 186 174 192 139 107 115 133 151 151 163 204 180 121 180 145 157 198 999
999 204 212 180 127  99  66  82  25  58  82  82 133 151 115 163 139 127 198 999
999  59 174 127  99 145 133 127  58  33  50  90 115 515 127 174 107  90 151 999
999 127  99  58 133 180 139 151 157 121 115 151 168 198 212 214 174 168 204 999
999 163 186 145 127 151 192 213 215 210 157 168 198 210 217 221 210 198 214 999
999 180 218 145  66 204 215 139 127 180 180 133 115  74 151 212 133 151 180 999
999 192 215 174 192 216 145   0   0  58 204 174 127  58 145 220 127 115 157 999
999  22 168 214 224 186   9   0   0   0 213 225 227 223 221 225 157 107 174 999
999   0  90 168 215 145  13   0   0   0 192 215 215 192 215 223 151 127 192 999
999   0  90 127 186 133  19   9   0   0 163 145  33   6 145 215 133 139 174 999
999   0  33 145 180 163   0   9   0   0 192 139 127 107 151 215 139 139 192 999
999  16 151 215 216 180   0   3   0   0 212 219 225 223 221 218  90  99 174 999
999 186 215 168 192 213  82   6   0  58 215 215 212 198 215 219  74  25 151 999
999 192 133   0  22 198 180  25  25 151 139  41  25  74 168 216 115  74 145 999
999 204 107  74  74  16 127  99  82  50  25  25  82 168 168 211 133 163 204 999
999 198 121 204 174  22  66  74  25  16  82 157 168 151 180 168  74 163 213 999
999 115 139 186 204 133 133 145 121 121 163 174 139 121 151 139  74 107 163 999
999 127 145 115 157 139 192 204 192 168 168 145 145 121 157 168 127 127 151 999
999 145 157 151 168 168 186 186 186 174 127  99 157 192 157 133 121 115 121 999
999 163 157 145 133 133 168 198 198 186 145  82 163 174 139  99 121 115 121 999
999 163 157 145 163 168 168 186 163 163 121  50 151 180  99  74 127  99 139 999
999 168 139 121 192 210 139 121 121 157 174  66  66  66  41 127 157  99 121 999
999 157 145 174 192 127  58  41 133 212 214 133  90  90  58 145 186  90  66 999
999 151 186 145  82  41 107 151 163 157 157 151 163 139  90 127 133  41  22 999
999 145 107  41  74 139 163 151 163 121 121 127 163 163 127 115  90 127 127 999
999  99  82 127 174 157 121 107 145 133 133 121  82  74 115 115  99 115 127 999
999 115 168 157 127 145 151 174 127 107 107  41  25  25  16   6   3   9  19 999
999 168 168 127 139 180 174 157  74  13  13   0   0   0   0   0   0   0   0 999
999 145 145 180 186 139  99  22   0   0   0   0   0   0   0   0   0   0   0 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
```

Channel 2

```
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 208 208 183 183 177 194 206 206 188 194 155 111 127 116 122 155 116 111 999
999 105 116 127 127 116 116 138 161 149 138 111  39  67  96 116 144 111 138 999
999  20  20  39  67  58  29  10  96  96  29  39  20   9  29  67  96  77 149 999
999 111  77  77  86  29  39  67 105  58   9  20  10   9  39  86  67  77 177 999
999 155 122  86  29  10   9  67  86  58  20  29   9  58  96  77  67  86 161 999
999 155 161 144  20   9  48  96  77  39  39  67 105 116  77  58  67  77 133 999
999  96 161 177 116  67  86 105 138  39  67 116 122 105   9  77 111  58 144 999
999  48 122 199 183 105 105 122 138  67 116 138  86  39  10  86 127 111 149 999
999  39   7 138 206 155 155 194 206 194 138  77  58  39  48  77  96  86 138 999
999  20   7  77 210 210 205 211 214 213 133  20  67  67  48  48  67 116 133 999
999   7  48  48 209 210 207 212 214 211  86  67  58  39  39  58  48  96 105 999
999  39  29 111 210 212 208 216 216 213 133  67  58  29  10  58  67 122  96 999
999  48  10 155 212 205 166 208 208 194 133  96  48   8  20  77  96 116 111 999
999   6  77 206 194  86  86 133 127  67 105 116  48   6   9  58  86 116 111 999
999  48 172 205 122  58  67 116 111  29  48  96  67  39  20  58  77 122 105 999
999 172 183  77  48  58  58  67  96  67  29  39  96 111  77  58  77 105  77 999
999 155  86  20  48  39  20  58  96  58  29   9  20  96 105  96  96 105  86 999
999  67  29  39  39  10  20  48  86  67  58  20   9  58  86 122 116 105  96 999
999  48  48  10  29  29  39  58  86  96  58  20  29  48  67 116 133 116  96 999
999 105  48  29  29  10  10  29  77  86  77  67  86  67  10  96 122 133 122 999
999 105 133 105  39  29  10  58  48  48  29  48  58  58   8  86 111 105 127 999
999  96 138 105  67  96  67  58  48  29  48  48  58  67  48 111 111 105 122 999
999  77  67  39  96 122  58  48  48  29  39  48  67  58 116 172 188 183 183 999
999  86 105  48  10  58 111 133 161 127  58  10  29  29 111 194 205 199 208 999
999 116 166  77   7  96 183 166 188 194 111   9   7   6  20 133 122 138 138 999
999 155 206 166 138 177 161 105 116 194 199  96  20   6  67 188 122 105  58 999
999 105 199 210 212 177  67  67  48 138 206 155 133 116 155 208 155 105 105 999
999   5 155 183 206 133  58 116  29 127 207 133  67   7  67 183 133 122 105 999
999   9 188 149 172 122  58 105 105 144 194  77   2   0   7 122  86 122 105 999
999   5 133 194 183 155   7  96  77 116 194  67  10   8  39 149 116 133 105 999
999  39 172 209 209 188   5  67  39 111 207 183 161 144 183 206 122 122  86 999
999 111 172 144 166 199 116  58  29 133 206 166 116  77 138 199 105  67  67 999
999 138 105   3   8 155 161  58  77 155 133  20   5   6  58 149  96  58  96 999
999 133  86  58  58  10 105  96  86  48  20   9  48  96 105 155 138 155 161 999
999  96  58 116 105  10  39  48  29  10  39  86 111 105 138 138 105 172 206 999
999   9   9  20  86  39  48  67  67  86  96  77  29  10  96 111 105 122 166 999
999   7   7   5   9  10  77  86  96  77  58   9   9   9  67 111 122 122 122 999
999   9   7   8   9   9  39  67  58  39   9   6  39  86  77  86  96 105  86 999
999   6   8   7   8   8  10  58  48  39  10   9  67 111  96 105  96 116  96 999
999   7   7   7   8   8  10  39  29  29  10   8  86 122  96  96 111 116 133 999
999  10   7   6  10  39  20  10  20  39  58  10  20  48  67 127 138 105 127 999
999  58  20  10  29  29   8   8  58 116 111  39  10  48  39 122 161  96  77 999
999  77 105  39   9  10  29  77  58  58  58  86  96  77  39  77 105  29  29 999
999 122  86   9   8  67  86  67  48  67  67 116 144 144 116  96 105 116 122 999
999  96  58  67  86  67  67  58  77 122 127 155 166 172 194 172 161 155 161 999
999  58  67  58  67  77 111 155 138 149 155 138 149 155 138 138 133 127 166 999
999 105  86  58 116 138 155 161 144 144 105  29  20  39   9  10  58  48  48 999
999 122 111 144 183 149 138  77  48  67   9  29  58  96  10  48  58  48  29 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
```

Channel 3

```
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 207 206 201 200 203 202 205 204 194 203 183 122 133 128 139 178 133 122 999
999 106 117 139 139 139 117 128 161 167 150 128  55  82 100 117 161 122 139 999
999  10  10  19  55  46  28  19 100  91  19  28  28   9  19  82 111  91 167 999
999 100  82  82  64  64  46  73 117  55   9  37  10   8  64 111  91  82 194 999
999 167 128  82  28  19  28  73 205  64  19  28  19  64 111  82 100 106 178 999
999 167 189 139   9  10  55 100  82  46  55  64 111 128  82  55 100  91 139 999
999 117 201 201 128  64  91 122 133  46  82 133 128 106   8  82 117  82 133 999
999 100 167 205 201  91  91 133 144  73 122 150 111  46   9 100 117 117 161 999
999  64   8 161 206 167 172 201 204 200 150 106  55  55  55  82 111 106 133 999
999  73   8  91 210 207 203 209 212 208 122  28  64  73  55  73  73 117 139 999
999  37  91  91 210 208 205 209 209 208  82  91  91  55  46  55  55 106  91 999
999  82  55 122 209 207 204 211 211 208 133  91  73  37  28  73  73 122 100 999
999 122  46 172 207 183 156 202 203 200 144 106  55   8  19 100 106 133 122 999
999  10 117 202 183  82  55 122 122  64 117 122  64   8  10  82 100 139 122 999
999  73 201 203 111  37  37 100 111  19  46  91 100  46  37  73  73 122 117 999
999 201 201  73  46  46  46  73 117  64  28  55 122 144  91  73 106 111  91 999
999 183  91   8  46  37  19  55 111  46  37  10  19 117 111 100 106 100 100 999
999  55  19   9  19   9  19  46  91  73  46  10  10  46 106 111 117 117 111 999
999  46  37  10  28  28  55  64  91 106  82  64  64  55  55 122 139 139 111 999
999 100  55  46  64  37  19  37  64  91  82  64 106  82  19 111 128 139 144 999
999 111 139 106  46  28  19  73  46  64  55  64  73  73  10 100 106 122 150 999
999  73 133 106  91 100  73  73  28  37  46  73  64  82  64 122 111 111 150 999
999  91  64  55 111 117  55  55  55  28  37  46  73  91 128 183 194 201 202 999
999 106 106  64  19  46 117 161 183 133  46  28  37  64 117 200 201 203 205 999
999 117 172  82   6 111 202 201 202 201 111   9   8   6  28 144 122 133 144 999
999 172 205 167 156 201 172 106 133 204 203 106  37   8  82 189 117 100  73 999
999 111 203 208 212 201  73 106  73 183 209 178 156 133 172 207 156 106 106 999
999   6 183 201 205 150  82 158  73 161 209 156  91   8  73 189 133 122 128 999
999  10 201 150 161 128  73 128 133 161 201  82   2   0   8 117 100 139 117 999
999   5 122 178 183 183  19 133 111 122 203  82  28   8  37 167 122 150 128 999
999  46 172 204 205 201  19 104  91 117 206 201 183 167 183 205 139 133 117 999
999 128 183 139 178 205 161  82  55 161 206 189 133 100 150 201 111  82 111 999
999 156  91   2   7 156 194 100 111 194 150  19   5   7  73 167  91  82 106 999
999 150  82  46  46  10 128 117 100  55  37   8  46 111 100 161 139 172 178 999
999 111  55 111 106   8  37  55  28  10  64 117 111 111 144 156 117 189 205 999
999   8   9  37  82  46  46  54  64 100 106 100  37  19  82 117 111 133 183 999
999   8   9   6   8  37 100 111 106  82  55  28  28  10  55 122 117 139 139 999
999  28   9   9  19  28  64  55  55  37  10   8  37 106  91 106 100 111 100 999
999  10   9   8   8   7  28  55  73  64  19   8  91 117 106 100 111 128 111 999
999  28   9   8  37  46  28  28  37  64  82  19  28  55  82 144 161 122 133 999
999  73  46  37  55  19   9   9  82 144 139  64  28  55  46 133 172 111 100 999
999 111 122  55   9  19  55 110  82  73  82 106 111 100  37  91 100  28  10 999
999 133  91   9  28 100 100  73  82  82  73 122 156 156 122 111  82 117 128 999
999 100  82  91 100  82  82  14  82 122 133 178 183 194 205 201 172 161 178 999
999  91 117  91  55  91 111 112 156 172 189 167 189 178 183 167 150 178 200 999
999 122 117 100 133 156 167 187 161 156 122  64  64  64  10  28  73  64  73 999
999 128 128 156 194 161 156  91  73  73  19  46  82 111  28 100  91  82  64 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
```

SOURCE: Amrine (1992). Reprinted with permission.

Table 6.6 CLASSIFICATION OF WATER IN CHANNEL 1

Gray value	Pixel count		Classification accuracy
	Water areas	Total	(%)
0	57	58	98
3	2	2	100
6	3	4	75
9	4	4	100
13	2	2	100
16	3	5	60
19	2	2	100
22	3	6	50

SOURCE: Amrine (1992). Reprinted with permission.

water shows why the water is not as spectrally distinct as in channel 1. There is a marked decrease in the overall accuracy when compared to channel 1. In channel 3, the water regions are again spectrally non-distinct. In fact, the problem in locating the water is similar to channel 2 but worse. Any pixels within the 5–200 gray range are labeled as water. Classification accuracy of the water subregions with channel 3 spectral data alone is very low. In fact, channels 2 and 3 are better equipped to pick up land cover types other than water, particularly pavement, when one lays Figure 6.32 and Figure 6.33 side-by-side.

B. Image Classification

Once the gray value ranges are located, we proceed to identify the bodies of water in the image. The BW classification model was extensively modified for this application. An objective function that will work to combine channels p and q is

$$\text{Max} \sum_{k=1}^{K} \left[\lambda_p' \sum_{i=1}^{n'} c_{ip}\, x_{ik} + \lambda_q' \sum_{i=1}^{n'} c_{iq}\, x_{ik} \right] \tag{6.25}$$

The size and border-length constraints as specified by the multiple subregion BW model were used (Equations 6.19 to 6.24). However, major improvements can be made to the model. These improvements include the pixel bounds constraint and the multi-criteria functions. The constraint that sets the spectral bound for each channel also sets the value of x_{ik} to zero if the pixel gray-value is out of this range. With these constraints, only the water-type pixels are considered for selection into a water subregion. For channel 1, these pixel-bound constraints look like:

$$\begin{aligned} c_{i1}\, x_{ik} &\le 22 \qquad i = 1, \ldots, K \\ x_{jk} &= 0 \qquad j \ne i,\, k = 1, \ldots, K \end{aligned} \tag{6.26}$$

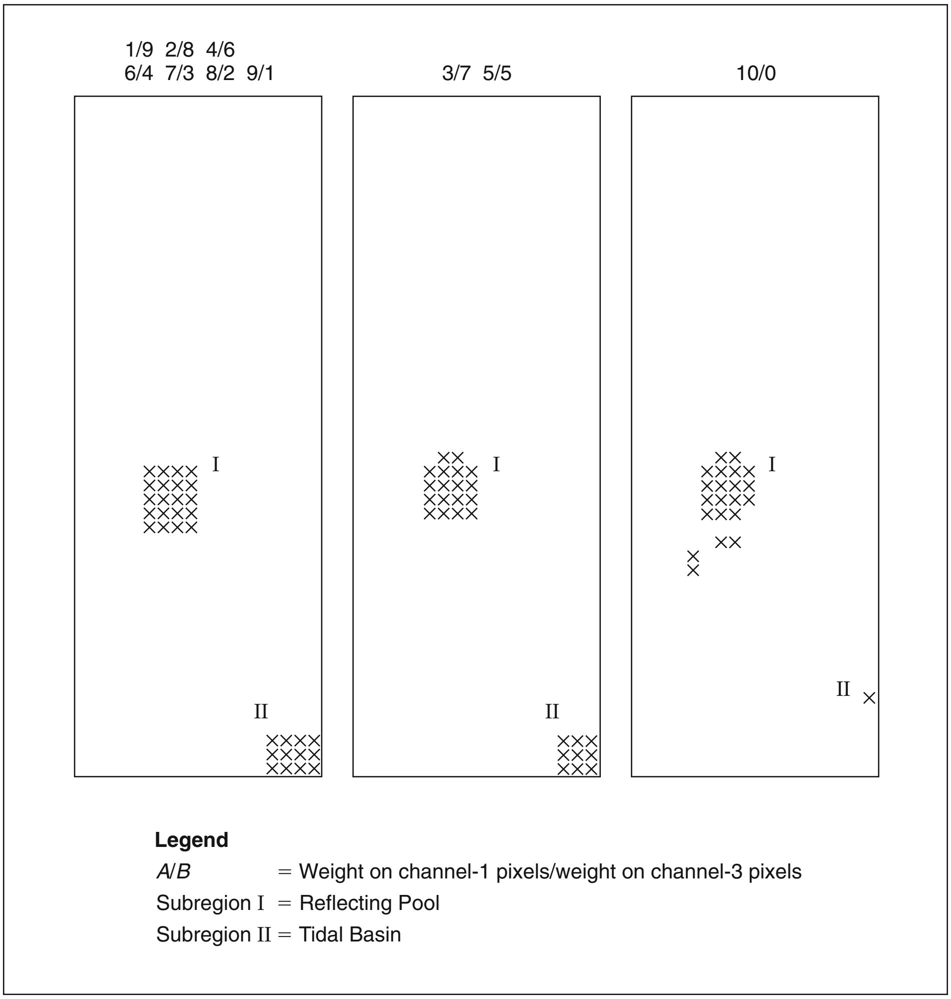
For channel 2:

$$\begin{aligned} c_{i2}\, x_{ik} &\geq 5 && i = 1, \ldots, n'; k = 1, \ldots, K \\ c_{i3}\, x_{jk} &\leq 166 && i = 1, \ldots, n'; k = 1, \ldots, K \\ x_{jk} &= 0 && j \neq i; K = 1, \ldots, K \end{aligned} \tag{6.27}$$

For channel 3:

$$\begin{aligned} c_{3}\, x_{ik} &\leq 5 && i = 1, \ldots, n'; k = 1, \ldots, K \\ c_{3}\, x_{jk} &\leq 200 && i = 1, \ldots, n'; k = 1, \ldots, K \\ x_{jk} &= 0 && j \neq i; k = 1, \ldots, K \end{aligned} \tag{6.28}$$

Figure 6.34 RESULTS OF RUNS FOR AREA ≥ 24 AND BORDER LENGTH ≤ 64


1/9 2/8 4/6
6/4 7/3 8/2 9/1
3/7 5/5
10/0
I
II
I
II
I
II
Legend
A/B = Weight on channel-1 pixels/weight on channel-3 pixels
Subregion I = Reflecting Pool
Subregion II = Tidal Basin

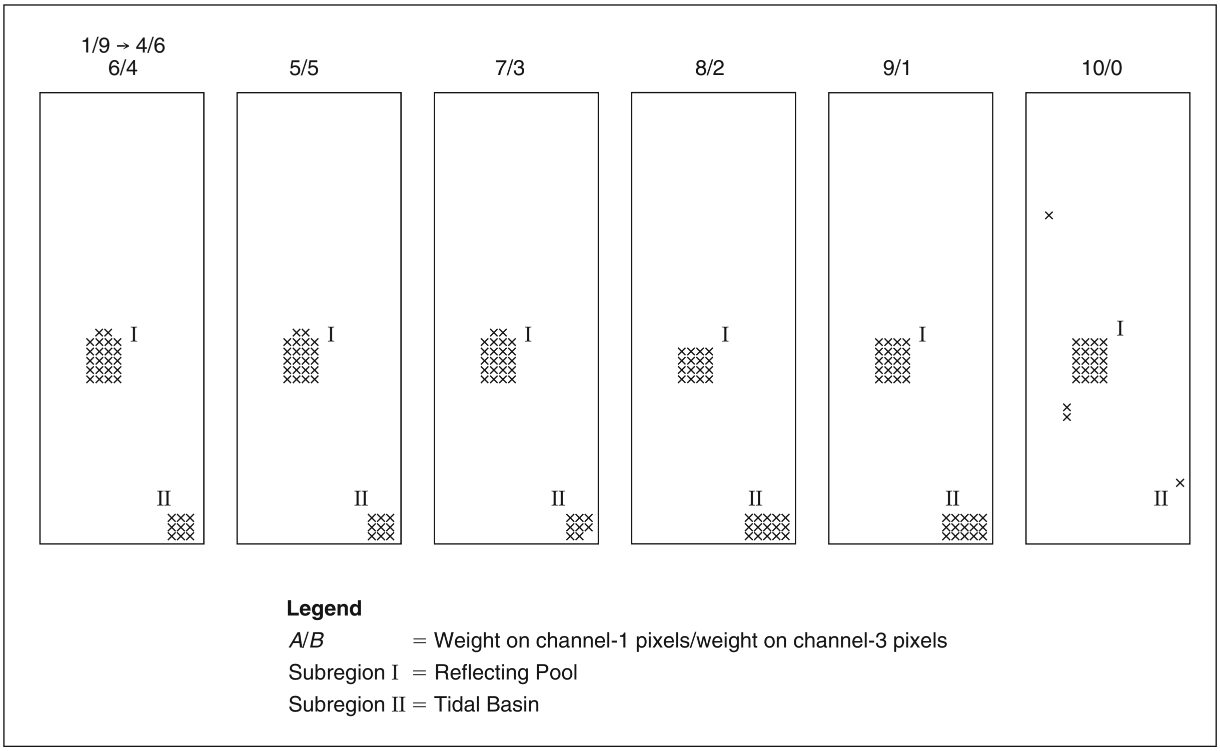
The area and border length constraints are also simplified. Instead of specifying an individual area and border length for each subregion, a total area and border length for all the subregions are specified:

$$\begin{aligned} &\sum_{k=1}^{K}\sum_{i=1}^{n'} x_{ik} \geq M \\ &\sum_{k=1}^{K}\sum_{i=1}^{n'}\sum_{j\in T_i} s_{ij}\,(P_{ijk} + N_{ijk}) \leq L \end{aligned} \qquad (6.29)$$

Multicriteria optimization is performed using the constraint-reduced feasible-region method[19]. The area is set parametrically at greater than or equal to 24 and the border length restricted to less than or equal to 64. Given these parameters, the modified BW model was run based on the tradeoff of information between channels 1 and 3. In this set of runs, the pixels selected are from the Reflection Pool and the Tidal Basin. The only exception to this statement is the run corresponding to the weights (10/0), where channel-1 is weighted by 10 and channel-3 weighted by 0 (in a scale of 10). In this run the selected pixels also come from the Constitution Gardens Lake and the noise-type pixels. Notice in all the solutions, the model uses the full border length limit of 64. However, the number of pixels selected varies from 24 to 31. Figure 6.34 depicts the results of the complete set of runs.

In a second set of runs, the area has been changed to be greater than or equal to 26 and the border length remains at less than or equal to 64. The model maximizes the objective function by selecting pixels from the Reflection Pool,

Figure 6.35 RESULTS OF RUNS FOR AREA ≥ 26 AND BORDER LENGTH ≤ 64


1/9 → 4/6
6/4
5/5
7/3
8/2
9/1
10/0
I
II
Legend
A/B = Weight on channel-1 pixels/weight on channel-3 pixels
Subregion I = Reflecting Pool
Subregion II = Tidal Basin

noise-type pixels, and the Tidal Basin. It is interesting to note that the size of the area selected varied from 26 to 32 pixels, but the border length of 64 is maintained. Most of the solutions are the same, except for runs corresponding to weight combinations 7/3, 8/2, and 9/1. For the 10/0 run, the model only identifies the Reflection Pool and a number of noise pixels. These results are very similar to the previous set of runs, including the weight setting 10/0. All these results are plotted in Figure 6.35 for easy reference. Even though only the channel-1/channel-3 combination is discussed here, the same type of classification can be performed between channels 1 and 2.

C. Lessons Learned

The results are very clear. A combination of two channels has been shown to yield better classification than one single channel. In the two sets of runs above, the 0/10 weight setting corresponds to using only channel-3 while the 10/0 setting corresponds to using only channel-1. In the former setting, the runs never converge. The latter setting also yields unsatisfactory results in that the Tidal Basin is missed altogether, and noise is picked up. Granted that channel 1 is the most suitable of all three channels for identifying water. But the extra information afforded by channel 3 (and for that matter channel 2) will help in the classification, even though we tend to rely more on channel 1 as reflected by the weights. In fact the best classifications came from weighing channel 1 much more heavily than channel 3 as verified by both sets of runs.

An alternative way of classifying the digital image would be to combine two channels into a composite index such as the normalized vegetation index (NVI). Then the BW classification model would operate on a single set of pixels representing the NVI. Such an attempt was followed but to no avail, and the process stops after a futile calculation of the vegetation indices. There are plausible explanations for this shortfall. Recall that the vegetation indices (VI) were defined as the difference between near infra red and red reflectances: VI = (near-IR) − (red), and the NVI was defined as NVI = VI/[(near − IR) + (red)]. The variables in these equations represent the gray value of a pixel in the red and near-IR imaging bands. With SPOT imagery, the red band corresponds to channel 2 and the near-IR band corresponds to channel 3. In general both indices result in high values for vegetation areas due to their relatively high near-IR reflectance and low red reflectance. In contrast, clouds, water, and snow have negative values due to their larger visible reflectance than near-IR reflectance.

Computation of VI and NVI, however, failed to show any negative pixel values in the four major water subregions. Two individual noise-type water pixels did have a negative value. It is suspected that the preprocessing of data in SPOT may have caused this problem. A second explanation is that the vegetation indices were specifically developed for the NOAA AVHRR system and are not applicable to SPOT images. A third explanation for the gray values to be out of the normal range is the imaging conditions. This is a catch-all factor that considers illumination angle/time-of-day, moisture content of gravel and soil, and water. All of these factors can affect the gray values that are recorded for a scene. Thus our initial assumptions about the inherent quality of processed satellite-image data are not supported. This points to the importance of understanding digital image processing as a prerequisite for proper use of remote sensing and GIS.

Computational time for the BW model amounts to hours per run on extreme cases, although most were accomplished around 45 minutes using the

Generalized-Algebraic-Modeling-System/Zero-One-Optimization-Model (GAMS/ZOOM) on a VAX 8550 mainframe computer. Research is under way to find a more efficient solver for the BW model, including the use of network-with-side-constraints routines[20] (Reed 1991; Earl et al. 1992). Simpler model formulations were also attempted (Warrender, Sovaiko, and Chan 1992). Preliminary results look promising.

XI. REMOTE SENSING, GIS, AND SPATIAL ANALYSIS

To the extent that the earth's surface is constantly reflecting and emitting electromagnetic radiation, remote sensing devices such as satellites are capable of measuring this radiation rather accurately and in a timely fashion. The intensities of emissions vary for the different wavelengths of the electromagnetic spectrum. The spectral distribution, or spectral signature, depends upon several factors, of which the most important are surface conditions, type of land cover, temperature, biological activity, and the angle of incoming radiation. Satellites (or equivalent remote sensors) equipped with multispectral scanners are able to measure the intensity in several bands of the spectrum. Since the bands span from infrared to red, such scanners can see beyond the naked eye for a number of geological, urban planning, agricultural, forestry, cartographic, and environmental management applications.

Unfortunately, remote sensing is a new technology that has yet to be fully integrated into GISs. SPOT represents one of the commercial efforts in integrating remote sensing with GIS. An attempt is made to use SPOT images to update GISs, which are typically organized into vector databases. These vector databases consist of digital line graphs (DLGs), TIGER, and DIME files, which are often outdated. Through digital or photographic images, framed in standard USGS map sizes, the company claims that the remote sensing information can be ingested into any major vector/raster GIS or AutoCAD®system.

Among other uses, GIS has been viewed as an integrated information base for analysis. The way in which analysis is linked to the database can be performed in three different ways (Anselin and Getis 1992). One can: (a) fully integrate all spatial analysis within the GIS software; (b) construct models of spatial analysis that efficiently link with the GIS and effectively exploit the spatial information in the database; or (c) leave the GIS and spatial analysis as two separate entities and simply import and export data in a common format between the two.

The third approach ignores the distinctive characteristics of a spatial database for use in spatial analysis. Nevertheless, it seems to be the approach most common in practice, mainly due to the problems with proprietary data formats in commercial GIS and the limited facilities of often awkward macro languages. Examples of this strategy are the joint use of GRASS and S for exploratory data analysis, the combination of SPANS and SYSTAT to carry out stepwise regression, and the use of ARC/INFO and BMDP for logistic regression.

The second approach is similar to the so-called modular design in integrated regional modeling and consists of developing self-contained modules for various types of spatial analyses. These modules are then linked to the specific data structures used in a commercial GIS. They are thus not "generic," but limited to a particular combination of GIS and analysis technique. Most of these modules are written and compiled separately and access the data structure of the GIS by means of proprietary library functions. In general, the use of the GIS macro facilities is avoided, given its poor performance in terms of speed. Even though this second approach links a statistical package to a GIS, it is generally limited to simple descriptive measures, such as univariate measures of spatial association. This has been referred to as Applications Programming Interface (API).

Finally, the first strategy is basically non-existent, due to the lack of analytical capabilities in most commercial GIS, with partial exceptions in SPANS and TRANSCAD. It is most closely approximated by the idea behind a spatial analysis toolkit. To the extent that spatial analysis includes all of the traditional techniques, the determination of an unambiguous set of generic spatial analysis functions in a GIS is an important, yet still largely unresolved question. This philosophy of designing GIS is sometimes referred to as client-server architecture.

Looking toward the future, the first strategy can become very useful when eventually implemented. Densham and Rushton (1992) demonstrated that processing cost for the most accurate, heuristic, location-allocation algorithms can be drastically reduced by exploiting the spatial structure of location-allocation problems. The strategy used—preprocessing inter-point distance data as both candidate and demand strings and using them to update an allocation table—allows the solution of large problems (3000 nodes) in a microcomputer-based, interactive decision-making environment. More importantly, these strategies yield solution times that increase approximately linearly (rather than exponentially) with problem size.

Along the same line, Ding, Baveja, and Batta (1994) implemented a facility-location model in GIS. The model locates facilities in a Manhattan metric (l_1-metric)[21] where travel can only take place in the east-west and north-south direction. Furthermore, travel has to avoid such barriers as lakes or other geographic obstacles. Its principal result is that the search for candidate facility-locations can be restricted to a finite, easily identifiable set of points. An example can be found in the "Facility Location" chapter of Chan (2005). Most importantly, the authors found the implementation greatly streamlined by a GIS such as ARC/INFO, assisted by the availability of TIGER files. The Densham and Rushton algorithm described above was employed to perform the location-allocation steps once the candidate facility locations are identified.

According to Bennion and O'Neill (1994), GIS is a very useful tool for defining transportation analysis zones (TAZ). Somewhat parallel to the BW districting model discussed above, they outlined an approach to address homogeneity and shape criteria for developing TAZs. By homogeneity is meant population density, employment density, average income, and so forth. In other words, we wish to group areas of similar population, employment density, and income together. By the same token, we wish to avoid irregular and elongated shapes, and only aggregate adjacent (rather than noncontiguous) geographic units together to form a zone. A fuzzy c-varieties algorithm is offered as a substitute for thematic mapping to model the homogeneity criterion, while analysis of fractal dimensions is used to address shape and compactness criteria. The fuzzy approach explicitly subjects the delineation of zonal boundary to human judgement and hence the boundary is not rigidly mandated by a priori rules. Fractal dimensions are used here to quantify the relationship between the area and perimeter of a polygon—a feature readily compatible with the data structure of most GISs. Bennion and O'Neill discussed future implementation of these procedures for ARC/INFO and ATLAS GIS.

Tomlin (1991) summarizes the cartographic modeling principles that may underpin eventual implementation of an integrated GIS on a digital computer, combining data and problem solving under one roof. He outlines the major conventions, capabilities, and techniques associated with this particular approach. His proposal differs from the competing techniques of relational database and feature- or object-oriented programming. In the widely disseminated relational idiom, geographic entities (such as lines or areas) are explicitly characterized in terms of attributes (such as names or numbers) and are related to one another by way of relations (such as adjacency or inclusion).[22] These relations can also be characterized in terms of their own attributes, and they too can be associated with one another by way of additional relations. The same is true in the now popular feature- or object-oriented idiom. Here, however, primitive entities can be associated with one another not only in terms of relations but in terms of more complex entities as well.

To be distinguished from these approaches, the fundamental spatial entity in cartographic modeling is the location. Unlike the units of data in most relational and object-oriented systems, locations are not units of "what" but of "where." Although locations can be aggregated into set of lines, areas, and surface features, they remain the elemental units for which attributes are recorded. The cartographic modeling approach associates locations with one another not with declarative statements specifying selected relations but with new entities that are generated by applying selected functions. To interrelated entities that are comprised of multiple locations, each is first disaggregated into a set of individual locations. A function is then applied to these locations to generate new attributes that will ultimately be re-aggregated to characterize the original entities or to form entirely new ones.

From this perspective, a question such as "How far is this area from that area?" would be expressed as "What is the minimum distance between any location within this area and any location within that area?" or "What is the distance between the centroid of this area and the centroid of that area?" The fact that there are two interpretations of that initial question reflects the utility of this point of view. It is a view that becomes particularly useful in dealing with more complex spatial relationships such as narrowness, enclosure, spottiness, interspersion, striation[23], and so on. The near future of cartographic modeling will likely be one of both refinement and extension. This is not only true in terms of new software (e.g., the MapBox system) but also in terms of new techniques in areas such as three-dimensional modeling, spatial statistics, interpolation, error tracking, feature extraction, temporal dynamics, flow simulation, and so on. Interoperability standards will make it easier to view, pan, and query geographic images and maps on the web. Efforts are ongoing to merge spatial data with non-spatial data in a single database. Such standardization brings all the advantages of a relational database management system to spatial data.

In general, spatial analysis contributes toward an understanding of locations and feature attributes (Galati 2006). Spatial analysis harnesses this duplicity through the study of geographic-feature locations and shapes. Spatial analysis relies heavily on the first and most fundamental law of geography. Attributed to Tobler (1965), this law states that everything is related to everything else, but near things are more related than distant things. We have seen in "Chapter 2—Descriptive Tools" how the gravity model represents this concept. Meanwhile, spatial statistics suggests **spatial autocorrelation** as a formal way to measure the degree to which near and distant objects are related. Positive spatial autocorrelation occurs when features that are close in location are also similar in attributes. Negative spatial autocorrelation occurs when features that are close together in space are dissimilar in attributes. Zero autocorrelation occurs when attributes are independent of location. In more advanced applications, the time axis can be included in a similar way as distance. When the time axis is used in conjunction with the space axis, they exhibit both spatial and temporal properties (Chan 2005). In short, spatial statistics provide analytic methods for describing the spatial relationships between geographic features. Spatial analysis—be it the gravity model, autocorrelation, overlay, or surface analysis—is an advanced and flexible form of data analysis. In a broader context, remote-sensing imaging and GIS offer a platform for specific applications of location and feature attributes, not only in geographic data, but also in imagery as well.

XII. CONCLUDING REMARKS

Over the past decades, GIS, automated cartography, and computer-aided design (CAD) have frequently been confused, both in the relative applicability of each technology in various fields and in the direction of basic research. The respective data structures have little in common: the literature of GIS has made little reference to automated cartography or to CAD, and the whole topic has had little relevance to automatic cartography or to CAD, let alone remote sensing. Today the development of technology for digitizing and display has clearly benefited from the influence of a much larger market for CAD. The industry is slowly moving toward this confluence. Similarly, remote sensing developments have been divorced from GIS, with the former being worked on by those involved in space technology and the latter by people involved with databases. However, the connection between the two is quite obvious, and they can greatly benefit from one another. Adams, Vonderobe, and Russell (1992) proposed a scheme to integrate GIS, CAD, facility management, and project management into a facility delivery system centered around spatial data.

Tomlinson and Associates (1987) stated that GIS is a unique field with its own set of research problems, although the entire GIS community would probably not agree with this view. A GIS is a tool for manipulation and analysis of spatial data; it therefore stands in the same relationship to spatial analysis as standard statistical packages such as SAS and SPSS stand to statistical analysis. Perhaps the most useful way of looking at GIS is to treat it as the data merging phase of image rectification, enhancement, classification, and merging. This view integrates remote sensing and information organization efforts very nicely.

Once data are properly organized, it follows that the set of potential applications of GIS is enormous, and is not currently satisfied by any other type of software. Future developments in GIS will depend on better algorithms and data structures, and continuing improvements in hardware. But they also need research in spatial analysis, in the development of better methods of manipulation and analyzing spatial data, and toward a better understanding of the nature of spatial data themselves through such issues as generalization, accuracy, and error and the integration with remote-sensing technology. Thus the future development in GIS needs to be concentrated in three areas: data structures and algorithms, spatial analysis, and spatial statistics. Development of hardware will probably continue to be motivated by larger markets in computer graphics and CAD.

In general, successful data collection, storage and retrieval depend upon (a) clear definition of the problem at hand; (b) evaluation of various data collection procedures; (c) identification of the collection procedures appropriate to the task; and (d) determination of the data interpretation procedure to be employed. In any approach to applying remote sensing, not only must the right mix of data acquisition and data interpretation techniques be chosen, but the right mix of conventional data collection techniques and remote sensing must also be identified. GIS and remote sensing are tools best applied in concert with one another. Their integration permits the synthesis and display of virtually unlimited sources of types of physical and socioeconomic data—as long as they can be computer coded with reference to a common geographic base.

Remote sensing affords us the capability to literally see the invisible. From remote sensing's aerial or space vantage point, we can also begin to see components of the environment on an ecosystems basis, in that remote sensing data can transcend the cultural and political boundaries within which much of our current resource data are collected. Continuing facility location and land use studies require an efficient information system for storing the relevant data. To be effective, information systems must be carefully designed in conjunction with the tasks to be performed by a particular study team. The most important phase of the design of information systems is the identification of the end use of each item of information planned for collection.

Recent development goes one step further to make GIS mobile. Users can transfer GIS databases from a desktop PC to a field unit. In the field, data can be added and edited directly in the GIS. Users can then upload the current database file to their office PC by way of an Internet site. The software also provides support for an optional GPS. Employing encryption technology, the PC can receive open database-compliant formats by way of wireless links from mobile units, and then combine that data into a single database on the office server. Through hand-held mobile devices such as palmtop computers, software exists for users to indentify starting points or destinations to obtain maps and directions. Unlike an existing Internet search engine, it can use proximity to deliver search results, thus automatically providing data relative to the user's current location. This capability lends itself directly to **E-commerce** (or most recently, mobile-commerce: **M-commerce**), defined as selling goods and services on the Web through the exploitation of information technology (Ngai and Gunasekaran 2007). Thus a nearby restaurant can interest this user in a meal should he feel hungry. Conversely, this user can have a keen awareness of what is around his current geographic location.

XIII. EXERCISES

Self-Instructional Module: LINEAR PROGRAMMING PART 1: MODEL FORMULATION

(to be found on the attached CD/DVD)[24]

In general, a mathematical model is either *deterministic* or *probabilistic*. For example, the models and algorithms shown in the Graph-Optimization module are deterministic. On the other hand, the queuing model in the Probability Distribution module is probabilistic. A linear program in its basic form is a deterministic model, although probabilistic versions have evolved.

Linear programming is the simplest and most elegant of optimization procedures. There are some very convenient features that result from dealing with linear equations. Notice the word "programming" in linear programming does not necessarily mean an electronic computer program, but rather a set of procedures to arrive at a solution: an *algorithm*. The algorithm is tedious enough that computer programming is invariably required for other than the smallest "textbook" problems.

The development of linear programming proceeded quickly during World War II when large scale economic and military planning were needed. Linear programming typically deals with allocating *limited resources* (such as labor, time, machines etc.) between *competing activities* (such as deployment of a particular type of aircraft) and results in the best possible mix (say using an equal number of long range and medium range aircraft for the mission). George Dantzig was the person most responsible for developing the *simplex algorithm* in 1947 for solving linear-programming models.

Using linear programs for decision-making typically involves two steps:

1. the mathematical modeling of the process, and
2. the application of an algorithm to solve the mathematical model, arriving at a desirable solution or a set of feasible solutions.

This modeling module and the accompanying solution-algorithm module are constructed to serve as an introduction to linear programming. After completing this and the accompanying module the reader should be able to:

(a) use linear programming to model a real-world problem, where appropriate;

(b) use the simplex algorithm to solve a specific type of linear programming problem.

The first part of the two-part module introduces the modeling procedure, while the second part shows the simplex algorithm as a way to solve the model. The simplex algorithm is a very systematic numerical procedure. This is one of the reasons we introduce it last, after the less mathematically prepared reader has a chance to get acclimated to analytics through five other modules and other exercises in working through this textbook.

In this first part of the LP module, we will studiy various LP models. It can be seen that LPs are used frequently to configure processes, programs, and plans. These examples are used for the sole purpose of learning the basic techniques. In the current chapter, entitled "Remote Sensing and Geographic Information System," some illustrative spatial applications of analytics are presented. The best example is political districting, in which residents are grouped into districts for voting purposes. In the formulation and solution of the political districting model, multicriteria linear integer programming (MCLIP) is required. MCLIP is a linear program in which the variables are integer valued (including binary values), and there is more than one objective function. Thus one may wish to form districts that are contiguous, compact and share common characteristics. Based on these criteria, a residential neighborhood is assigned to a district if its binary variable is unitary valued, and not assigned otherwise.

Problem 1: Bayesian and Contextual Classification

A computer input file has been provided in book Figure 6.27 to execute the Benabdallah-and-Wright (B&W) model. It is reproduced in processable form for your convenience under this folder "CDtoBeBurned\Software&Data\Book".

Please perform the following tasks:

(a) Run the B&W Model on the sample data provided, using an available software such as LINGO. Depending on the optimization software, the given file may need to be edited.

(b) Discuss the results in terms of the non-inferior solution-set of a multi-criteria optimization model. (Please refer to book Figures 6.28 and 6.29 of the "Remote sensing & GIS" book chapter.)

While item 1 above is the prerequisite, item 2 is really the interesting part of this exercise, as you may agree.

Problem 2: TS-IP Image-Processing Software

Included in the book CD/DVD is the TS-IP image-processing software. This software is described in book Sections 6-V and 6-VI, and further explained in a User's Manual included in the CD/DVD. Figure 6.36 shows the screen capture of the software as it displays a hurricane approaching Cuba in the form of an unprocessed image. Also shown is a histogram of the gray values of the raw image.

Figure 6.37 shows the screen capture of a processed image, as well as the menu of the software.

(a) Familiarize yourself with the TS-IP software. Then try to reproduce the processed image as shown. In this regard, you might wish to consult book Exercise F in the "Exercises and Problems" book Chapter

(b) Can you replicate the histogram shown in Figure 6.38?

Figure 6.36 ORIGINAL CUBAN IMAGE

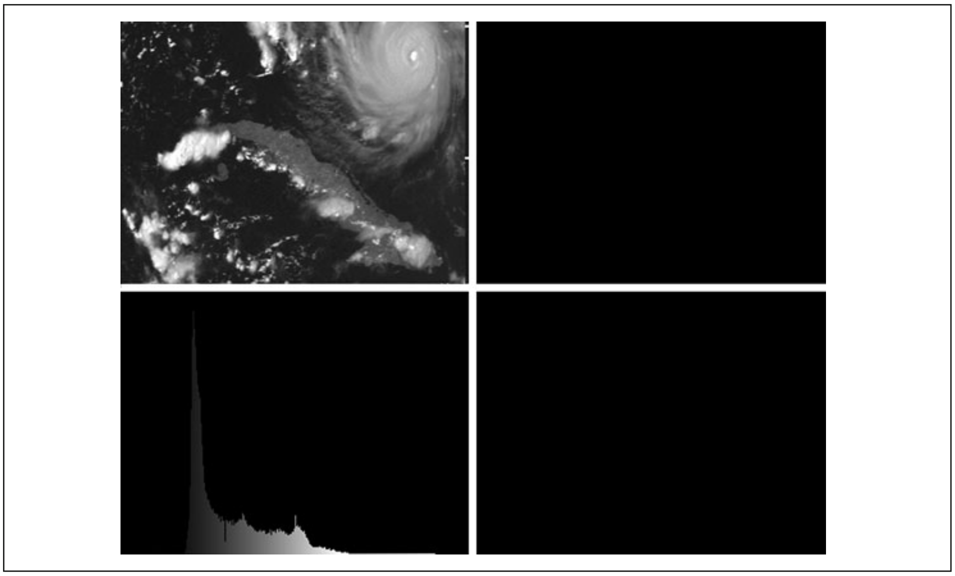


Figure 6.37 PROCESSED CUBAN IMAGE

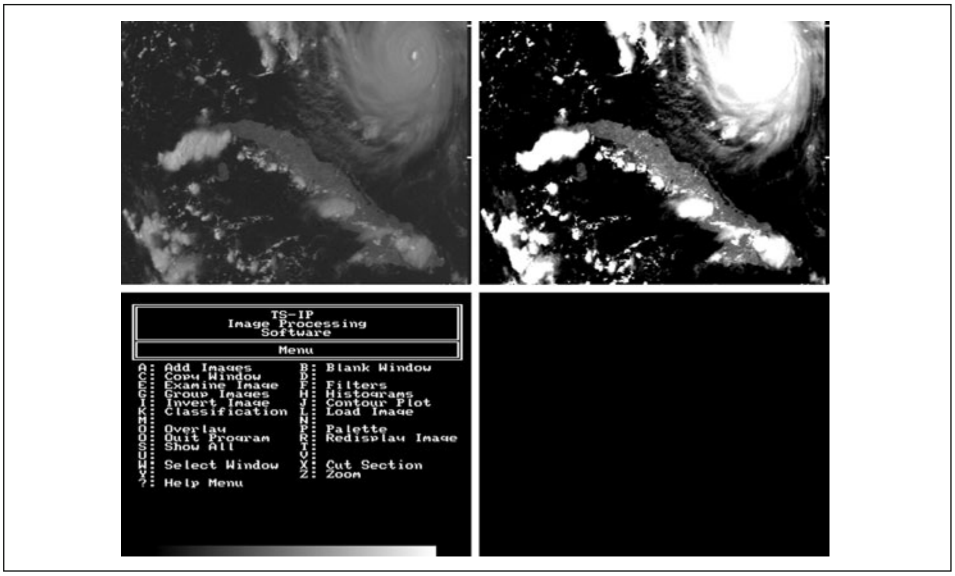
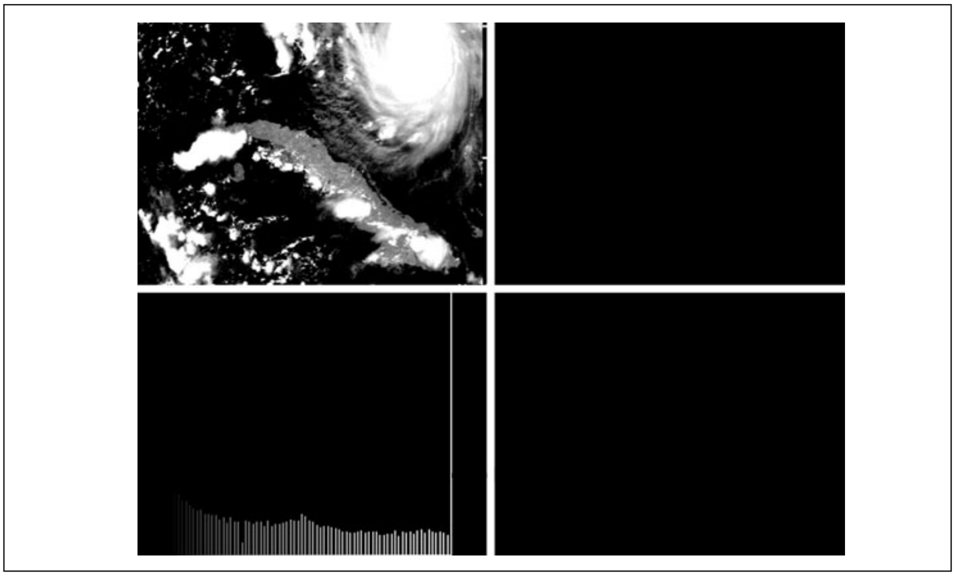


Figure 6.38 HISTOGRAM OF PROCESSED IMAGE



ENDNOTES

[1] See, for example, the "Spatial-Temporal Information" chapter of Chan (2005) under Veronoi diagrams.

[2] For an example, see the "Space Filling Curve" discussion in Chan (2005).

[3] The gravity model was discussed in Chapters 2 and 3.

[4] For a discussion of MAUT, refer to Chapter 5.

[5] Frequency, or how many times a sine wave "wiggles" in a period, is formally defined as the number of waves completed in 360-degree (2π radians) rotation of θ in $\sin\theta$ Thus the sine wave $\sin 2\pi$ has a frequency of one, and the sine wave of $\sin 4\pi$ has a frequency of two and so forth.

[6] Amplitude is the maximum and minimum absolute height of a sine wave. Thus the sine wave $\sin\theta$ has an amplitude of 1 and the sine wave $2\sin\theta$ has an aplitude of 2.

[7] Phase is the horizontal displacement of the sine or cosine wave. For the sine wave $\sin\theta$, for example, the phase is measured in the displacement quantity $\Delta\theta$.

[8] For the relationship between Kriging and Spatial Time Series, see Chan (2005) under the latter.

[9] For an explanation of the Sobel operator, see section VI-B of this chapter. The Sobel filter is a way to detect edges or lines in an image.

[10] For an explanation of the fast Fourier transform, consult Section VI-A of this chapter. The Fourier transform examines the data in its frequency (spectral) domain in order to detect noise. A filter is then applied to remove the noise.

[11] As explained the section VI-A of this chapter, the median filter is used to correct striping and snowy images.

[12] For an explanation of posterior probabilities and the Bayesian classifier, see the "Bayesian Decision Making" section of Chapter 3.

[13] For a review of Bayes' theorem, please refer to Chapter 3 under the "Bayesian Decision Making" section.

[14] For a formal definition of a non-inferior solution (or the analogous terms of a non-dominated solution or efficient frontier), see Chapter 5.

[15] As explained in Chapter 5 under "Exploring the Efficient Frontier" section, this is refered to as the constraint-reduced feasible-region procedure.

[16] Between the two sets of binary decision variables P_{ij} and N_{ij}, each segment s_{ij} is counted twice, and the resulting border length is recorded as twice the actual value.

[17] The entropy function was introduced in Chapter 3.

[18] The TS-IP imaging software was explained in Section VII of this chapter and is included on a CD-ROM at the back of this book.

[19] For an explaination of the constraint-reduced feasible-region method, see Chapter 5 under the "Exploring the Efficient Frontier" section.

[20] For an introduction to network-with-side-constraints, see Appendix 4

[21] For further explanation of l_1-metric, see Chapter 5 under "Goal setting."

[22] For an example of relational database, see Section III-A of this chapter.

[23] Marking with stripes.

[24] The answer to this Module is attached at the end of this text book.

REFERENCES

Adams, T. M.; Vonderohe, A. P.; Russell, J. S. (1992). "Integrating Facility Delivery through Spatial Information." *Journal of Urban Planning and Development* 118, No. 1:13–23.

Ahituv, N.; Berman, O. (1988). *Operations Management of Distributed Service Networks: A Practical Quantitative Approach*. New York: Plenum Press.

Amato, I. (1999). "God's Eyes for Sale." *Technology Review* (March-April):36–41.

American Society of Civil Engineers (1996). "Mapping the Future." *Civil Engineering* (July):18–19.

Amrine, J. M. (1992). Spectral and spatial pattern recognition in digital imagery. Master's Thesis. (AFIT/GSO/ENS/92D-01). Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson AFB, Ohio.

Anderson, J. R.; Hardy, E. E.; Roach, J. R.; Witmer, R. E. (1976). *A land-use and land cover classification system for use with remote sensor data*. Washington, D.C.: U. S. Geological Survey.

Anselin, L.; Getis, A. (1992). "Spatial statistical analysis and geographic information systems." *The Annals of Regional Science* 26:19–33.

Benabdallah, S.; Wright, J. R. (1992). "Multiple subregion allocation models." *Journal of Urban Planning and Development* 118, No. 1:24–40.

Bennion, M. W.; O'Neill, W. (1994). Building transportation analysis zones using GIS. (Presentation Paper 94-0476). Paper presented at the 73rd Annual Meeting of the Transportation Research Board, Washington, D. C.

Besag, J. (1989). "Digital image processing: Toward Bayesian image analysis." *Journal of Applied Statistics* 16, No. 3:395–407.

Brigantic, R.; Chan, Y. (1994). A comparison and contrast of Bayesian classification vs. the Benabdallah and Wright procedure of spatial pattern Recognition." Working Paper. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson AFB, Ohio.

Bushenkov, V. A.; Chernykh, O. L.; Kamenev, G. K.; Lotov, A. V. (1994). "Multidimensional images given by mappings: Construction and visualization." *Pattern Recognition and Image Analysis* 5:35–56.

Chan, Y. (2005). *Location, transport and land-use: Modeling spatial-temporal information.* Berlin and New York: Springer.

Conner, P. K.; Mooneyhan, D. W. (1985). "Practical applications of LANDSAT Data." In *Monitoring earth, ocean, land, and atmosphere from space.* (Progress in astronautics and aeronautics, Vol 97), edited by A. Schnapf. Washington, D.C.: American Institute of Aeronautics and Astronautics, 371–396.

Corbey, K. P. (1996). "One-meter satellites: practical applications by spatial data users." *Geo Info Systems* (Supplement) (July):39–42.

Cressie, N. (1991). *Statistics for spatial data*. New York: Wiley-Interscience.

Densham, P. J.; Rushton, G. (1992). "Strategies for solving large location-allocation problems by heuristic methods." *Environment and Planning A* 24:289–304.

Ding, Y.; Baveja, A.; Batta, R. (1994). "Implementing Larson and Sadiq's location model in a geographic information system." *Computers and Operations Research* 21, No. 4:447–454.

Earl, A. J.; McGuiness, J. J.; Chan, Y. (1992). Pixel subregion allocation. Working Paper. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson AFB, Ohio.

Eddy, C. A.; Looney, B. (1993). "Three-dimensional digital imaging of environmental data: Selection of gridding parameters." *International Journal of Geographic Information Systems* 7:165–172.

Ehlers, M. (1995). "Integrating remote sensing and GIS for environmental monitoring and modeling: Where are we?" *Geo Info Systems* (July):36–43.

Engelhart, J. (2000). "What's E-commerce have to do with GIS." *Geo Info Systems* (January):58.

Feng, C.; Wei, H.; Lee, J. (1999). WWW-GIS strategies for transportation applications. Paper presented at the 78th annual meeting of the Transportation Research Board, Washington, D. C.

Fischer, M. M.; Nijkamp, P., eds. (1993). *Geographic information systems, spatial modelling, and policy evaluation*. Berlin, Germany: Springer-Verlag.

Foley, T. M. (1994). "Zooming in on remote sensing markets." *Aerospace America* (October):22–27.

Galati, S. R. (2006). *Geographic Information Systems Demystified*. Boston and London: Artech House.

Gonzalez, R. C.; Woods, R. E. (1992). *Digital image processing*. Reading, Massachusetts: Addison-Wesley.

Green, D.; Chan, Y. (1994). "Computational Aspects of the Benabdallah-and-Wright Districting Model." Working Paper. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson AFB, Ohio.

Gryder, R. K. (1992). TIGER5-Extraction of geographic information format from the TIGER System. (Report ORNL/TM-12061). Oak Ridge National Laboratory, Martin Marietta. Oak Ridge, Tennessee.

Gualtieri, G.; Tartaglia, M. (1998). "Predicting urban traffic air pollution: A GIS framework." *Transportation Research D* 3, No. 5:329–336.

Heacock, E. L. (1985). "US remote sensing of the earth from space—A look ahead." In *Monitoring earth, ocean, land, and atmosphere from space*. (Progress in astronautics and aeronautics, Vol 97), edited by A. Schnapf. Washington, D.C.: American Institute of Aeronautics and Astronautics, 713–745.

Heagerty, P. J.; Lele, S. R. "A composite likelihood approach to binary spatial data." *Journal for the American Statistical Association* 93, No. 443:1099–1111.

Horowitz, A. J. (1997). "Integrating GIS concepts into transportation network data structures." *Transportation Planning and Technology* 21:139–153.

Huang, Z.; Shin, K. G. (1996). "A new location coding scheme for intelligent transportation systems." *ITS Journal* 3, No. 2:99–109.

Hutchinson, B. G. (1974). *Principles of urban transport systems planning*. New York: McGraw-Hill.

Jusoff, K.; Hassan, H. M. (1998). "An overview of satellite remote sensing for land use planning with special emphasis on Malaysia." *Remote Sensing Review* 16:209–231.

Kelso, T. S.; Chan, Y.; Ursi, R.; Smith, B. (1995). TS-IP Users Guide—Version 2.8 Department of Operational Sciences. Airforce Institute of Technology. Wright-Patterson AFB, Ohio.

Klosterman, R. E. (1991). *TIGER: A primer for planners,* (Planning Advisory Service Report 436). Chicago, Illinois: American Planning Association.

Koch, T. (1999). "GIS: Mapping the OR/MS world." *OR/MS Today* (August):26–30.

Langran, G. (1992). *Time in geographic information systems.* London: Taylor and Francis.

Lazar, B. (1996). "Understanding SDTS topological vector profile implementation." *Geo Info Systems* (June):42–45.

Lee, Y. C.; Zhang, G. Y. (1989). "Development of geographic information systems technology." *Journal of Surveying Engineering* 115, No. 3:304–323.

Leung, Y. (1997). *Intelligent spatial decision support systems.* Berlin, Germany: Springer-Verlag.

Lillesand, M. T.; Kiefer, R. W. (1987). *Remote sensing and image interpretation.* New York: Wiley.

Longley, P.; Batty, M., eds. (1996). Spatial modelling and GIS. Cambridge, England: GeoInformation International.

Marble, D. F.; Peuquet, D. J. (1988). "Geographic information systems and remote sensing." In *Manual of remote sensing*, edited by D. S. Simonett. Bethesda, Maryland: American Society of Photogrammetry, 923–958.

McCord, M. R.; Jafar, F.; Merry, C. J. (1996). Estimated satellite coverage for traffic data collection. Working Paper. Department of Civil/Environmental Engineering and Geodetic Science. Ohio State University. Columbus, Ohio. .

McCrary, S. W.; Benjamin, C. O.; Ambavanekar, V. E. (1996). "Consensus building model to select OASIS in small communities." *Journal of Urban Planning and Development* 122, No. 2:46–70.

McLachlan, G. J. (1992). *Discriminant analysis and statistical pattern recognition*. New York: Wiley-Interscience.

Mehrotra, A.; Johnson, E. L. (1995). Taking the politics out of districting. Working Paper. Department of Management Science. University of Miami. Miami, Florida.

Narumalani, S.; Zhou, Y.; Jelinski, D. E. (1998). "Utilizing geometric attributes of spatial information to improve digital image classification." *Remote Sensing Reviews* 16: 233–253.

Ngai, E. W. T.; Gunasekaran, A. (2007). "A review for mobile commerce research and applications." *Decision Support Systems* 43:3–15.

Nyerges, T. L. (1991). "Geographic information abstractions: conceptual clarity for geographic modeling." *Environment and Planning A* 23:1483–1499.

Nyerges, T. L.; Dueker, K. J. (1988). Geographic information system in transportation. Report to the U.S. Department of Transportation. Federal Highway Administration. Washington, D. C.

O'Neill, W. A.; Harper, E. (1997). Location translation within a GIS. (Presentation Paper No. 97-1246). Paper presented at the 76th annual meeting of the Transportation Research Board, Washington, D. C.

Pace, P. J.; Evers, T. K. (1996). "Oak Ridge National Laboratory develops GISST data server." *Geo Info Systems* (November):32–39.

Reed, T. G. (1991). Binary programming models of spatial pattern recognition: Applications in remote sensing image analysis. Master's Thesis. Air Force Institute of Technology. Wright-Patterson AFB, Ohio.

Sabins, F. F. (1987). *Remote sensing: Principles and interpretation*, 2nd ed. San Francisco: Freeman.

Schweiger, C. L. (1992). Current use of geographic information systems in transit planning. (Report No. DOT-T-92-02). U. S. Department of Transportation. Washington, D. C.

Shaw, D. T.; Maidment, D. R.; Arimes, G. N. (1993). "SITE CODE: Computer-based regulatory information for site development." *Journal of Urban Planning and Development* 119, No. 1:1–14.

Shen, Y. (1995). "SIR-C advanced imaging radar studies the earth." *Aerospace America* (April):38–43.

Shih, S. F. (1988). "Satellite data and geographic information system for land use classification." *Journal of Irrigation and Drainage Engineering* 114, No. 3:505–519.

Star, J.; Estes, J. (1990). *Geographic information systems: An introduction.* Englewood Cliffs, New Jersey: Prentice Hall.

Szekielda, K-H. (1988). *Satellite monitoring of the earth*. New York: Wiley.

Tobler, W. R. (1965). "Computation of the correspondence of geographical patterns." *Papers and Proceedings of the Regional Science Association* 15:131–139.

Tomlin, C. D. (1991). "Cartographic modeling." In *Geographical information systems: Principles and applications*, edited by M. F. Goodchild, D. J. Mcguire, and D. W. Rhind. Longman Group Ltd.: Essex, England.

Tomlin, C. D. (1990). *Geographic information systems and cartographic modeling*. Englewood Cliffs, New Jersey: Prentice-Hall.

Tomlinson and Associates (1987). "Current and potential uses of geographic information systems—the North American experience." *International Journal of Geographical Information Systems* 1, No. 3:203–218.

Transportation Research Board (1984). "Census data and urban transportation planning in the 1980s." *Transportation Research Record* 981.

U. S. Geological Survey (1983). Map of Washington West, D. C.–MD.–VA., EE-000123. (38077-H1-TB-024).

Ware, R. (1986). Description of the land use suitability assessment implementation evaluation phase: process and procedure. Regional Planning and Coordinating Commission of Greene County, Ohio.

Warrender, C.; Sovaiko, S.; Chan, Y. (1992). Subregion allocation through optimization. Working Paper. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson AFB, Ohio.

Werner, P. A. (1974). A survey of national geocoding systems. (Report No. DOT-TSC-OST-74-26). U. S. Department of Transportation, Washington, D. C.

Wright, S. A. (1994). Private discussions.

Wright, S. A.; Chan, Y. (1994). Pure and polluted groundwater classification on a pixel amp. Working Paper. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson AFB, Ohio.

Wright, S. A.; Chan, Y. (1994a). Multicriteria decision-making applied to the iterated-conditional-modes contextual image classification technique." Working Paper. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson AFB, Ohio.

Zimmerman, P. (1988). "Photos from Space—Why Restrictions Won't Work." *Technology Review* (May–June):45–53.

Zhao, Y. (1997). *Vehicle location and navigation systems*. Norwood, Mass.: Artech House.

7

Analytics and Spatial Information Technology: Retrospect and Prospects

"We always overestimate the change that will occur in the next two years and underestimate the change that will occur in the next ten. Don't let yourself be lulled into inaction."
Bill Gates

Decision makers increasingly rely upon extracting relevant information from large repositories of data in order to make strategic decisions. The information is often used to test hypotheses or discover insights or intelligence (Chan et al. 2009). The support for strategic decisions is usually based upon data collected from internal operations, supplemented by relevant external information, and insights gained from a model of the real world. We logically associate all the tools, techniques, and processes associated with deriving intelligence from the core data as "Analytics." We include many diverse techniques within the broad term **analytics**, including statistics, predictive models, visualization systems, etc. The important point is that all these techniques may play a role in knowledge discovery and decision-making (Davenport and Harris 2007). Simply put, analytics is "the science of analysis." A more practical definition would be how an individual or enterprise arrives at sound decisions based on existing data. For the purpose of this book, the field of analytics includes the use of mathematical models, building upon statistics, probability, simulation, optimization, decision analysis, and most notably, spatial analysis. Analytics closely resembles data mining, but tends to be based on modeling, sometime involving extensive computation.

In this volume, we have presented a range of decision-making technologies to support facility location and land use. By no means are we offering a comprehensive treatise on analytics. Rather, we have geospatial applications

Y. Chan, *Location Theory and Decision Analysis*, 2nd ed., DOI 10.1007/978-3-642-15663-2_7,

in mind. To further this goal, we focus on spatial association, spatial clustering, geographic information system, and image processing among the above topics for analytics. Geospatial analysis is one of twelve fastest growing knowledge-based professional fields. According to the University of Pennsylvania, “Spatial analysts are helping retailers and service providers find store and facility locations, working with transit providers to provide real-time vehicle information, developing strategies for community policing, helping city planners promote infill development and combat urban sprawl, and working with professionals in many disciplines to explore the likely impacts of global climate change.” For that reason, universities have offered programs of study in this area. We hope that this book serves both the professionals as well as the classroom in disseminating and discover knowledge in this field.

I. ANALYTICS

Spatial analytics builds upon regular analytics, but goes well beyond, in that it handles the special, often complicated, features of spatial data. We can call this “ . . . the third segment of analytics consist[ing] of a set of more advanced analytical skills and methodologies . . . ” according to Bell (2008). The inclusion of spatial-data handling in mainstream database software has grown consistently and the emergence of a healthy open-source geospatial software community has meant that the mainstream and spatial database worlds have been converging. As spatial analytics receives increased attention, the distinction between basic analytics and spatial analytics becomes blurred. As a first step, let us review the fundamentals of analytics as a foundation for the more special field of spatial analytics.

As stated in the outset, a distinguishing feature of this book is to point out how various participating disciplines rely on the same pool of fundamental techniques to solve their respective problems. We already suggested “mainstream” analytics and spatial analytics are converging. We go further by examining the diverse disciplines that perform work in this area, ranging from engineers to planners, from geographers to political scientists, from economists to management scientists. We wish to show how different terminologies in these varied disciplines can be unified, since—unknown to even the experts in the respective fields—many have the same meaning. More important, they are built upon the same mathematical technique. We wish to review analytics with this goal in mind, and here are some examples.

A. Statistical Modeling

Take the very simple case where we want to establish a relationship between, say, employment and population. We want to show that an area with larger employment also has higher supporting population. In classical linear regression, we may wish to define population as the dependent variable, and employment as the independent variable. The dependent variable appears on the left-hand-side and the independent variables on the right-hand-side of the equation. While these are commonly adopted terminologies, some readers would know that in some quarters, independent variables are also called *explanatory variables* or *regressors*. At the same time, some would also know that the dependent variable is sometimes called *response variable*.

Let us go further in this example. In canonical correlation, one ascertains if a set of *criterion variables* is possibly affected by a set of *predictor (input) variables*. In econometrics, simultaneous equations are used to pose the relationship between a number of *endogenous variables* and *exogenous variables*. Endogenous variables appear on both the left-hand-side and right-hand-side of the equations, while exogenous variables only appear on the right-hand-side. Apparently, these two techniques—canonical correlation and econometrics—use the two sets of terms to accomplish a similar function. However, the author does not see too many occasions in which these parallels are pointed out. It was one of the motivations for the current volume to be compiled, to see that this area of analytics may have a common mathematical denominator.

B. Optimization

Let us take another example. In optimization problems, a figure-of-merit is usually maximized or minimized. For example, profit is to be maximized while cost is to be minimized. This figure-of-merit is expressed in terms of an objective function. Again, in some quarters, the objective function is referred to as a *functional* (See Appendix 1). In the management-science community, profit or cost is expressed in terms of a set of decision variables. In the engineering community, they like to think that the functional is driven by a number of *control variables*.

In short, the two terms—*objective function* and *functional*—are traditionally used in the respective disciplines when they perform optimization, but are somewhat equivalent. Both define a domain whose elements are functions, sets and the like, and a figure-of-merit is to be maximized or minimized. In the process, management scientists typically deal with cross-sectional data, while engineers often worry about the behavior of a system over time. Accordingly, the term *objective functional* refers to the integration of a functional over time, while objective functions are generally static or steady-state expressions referring to an average performance of a system.

An objective function satisfies its optimality conditions when it is optimized at a point (or points) within the feasible region as defined by the context of the problem. A continuously differentiable function is optimized when it has a relative maximum or minimum at a point assuming a "zero slope or gradient." This value lies at an interior point of the feasible region, including its boundary on special occasions. In the engineering community, the function is said to be *stationary* at the optimum. A stationary point is obtained by setting the gradient of the functional to zero. However, stationarity also includes an *inflexion point* of the function, which is not a local (or global) optimum.

C. Multicriteria Decision-Making

Utility theory is a key foundation of economics and operations research. The basic premise is that a number of disparate metrics can be translated into a common unit called *utiles*. Once this is done, cross comparison can then be made among alternatives with seemingly incommensurate attributes or criteria. This is generally accomplished by a multi-attribute utility-function, which—after valuation—combines the incommensurate attributes or criteria through weights and scaling constants. Once these incommensurate attributes are converted into a "common currency of exchange," the utiles, the preferred alternative(s) can then be picked among competing alternatives.

In Chapter 5, we pointed out that cross comparison among alternatives can still be possible without a multi-attribute utility function, although in a more limited sense. For example, a shirt that is cheaper and better quality is always preferred to one that is more expensive and inferior in quality. Here, no utility function needs to be constructed to combine price and quality—the two incommensurate attributes—into utiles before a decision can be made between them. And more importantly, there are occasions where multi-attribute utility theory may not apply, when one is required to use techniques other than utility function to rank order alternatives.

Before we go into spatial analytics, let us repeat our premise on general analytics. Depending on the occasion and the participating discipline, different technical terms are often used to convey similar ideas. While there are truly disciplinary differences in focus and concerns, there are similar analytical tools to solve seemingly diverse problems. When we found that there were very few places where such analytical parallels are established, we decided that this book—and a companion volume, Chan (2005)—may just fill part of that niche in the literature. Aside from the texts, we wish to draw the reader's attention to the Glossary of Technical Concepts in Appendix 5, in which we try to cut across the jargons used in different disciplines one at a time.

D. Location-Based Analysis

Another distinguishing feature of this book is that it explicitly considers geographic attributes, network effects, and interaction between economic and noneconomic activities between different subareas within a vicinity. We call it **spatial analytics**. Spatial analytics analyzes problems with full recognition of more than one dimension. For example, aspatial analysis deals only with aggregate attributes such as the total population and employment in the entire region, the total amount of retail floor space, the total acreage of parks and recreation areas, the total number of hospitals, and perhaps the aggregate economic growth over time. It does not disaggregate by zones or other subareal units, neither does it explicitly deal with interzonal interactions such as commuting between employment centers and population centers. Having to include this interaction, spatial analysis is multidimensional and generally more complex than aspatial analysis.

Let us illustrate with more examples. Entering a new community, an individual often worries about such decisions as where to find a home. A business would like to know where to locate a shop. These are often called facility-location problems. Urban planners are typically concerned with the former decision, while industrial engineers are usually concerned with the latter decision. A moment of reflection shows that these two seemingly disparate decisions are related. People like to live reasonably close to work, and business like to be proximal to the clients they wish to serve, not to say within acceptable commuting distance for their work force. Considering all the residents and all the businesses in the community, their cumulative locational decisions result in a land-use pattern, which determines the resultant economic growth and noneconomic impacts. This is now the concern of not only the urban planner or the industrial engineer, but also the rest of the community, including businesses, community leaders and politicians.

As with other enterprises, the ultimate goal of facility location and land use is to serve the client demands or the constituents. Thus a fire station is located for the sole purpose to put out people's fires quickly, while a city master plan aims to provide all the services to the local population with distinction. The way the clients are ultimately served reflects the merit or demerits of a spatial decision. This service pattern is often referred to as the *demand allocation* or *activity distribution*, depending on—once again—whether one is an industrial engineer or an urban planner. An industrial engineer allocates demands or workloads efficiently among workstations in a manufacturing plant by minimizing movements between related workstations. For the urban planner, a compact city form cuts down on the commuting. This way, the planner can cut down on the activity distribution around town, resulting in less traffic congestion and environment pollution.

Municipal service may be provided by a single facility or a combination of facilities. Thus, the population may depend only on one fire station in a small town, while it depends on many fire stations in a large city. Sometimes, specialized services may only be provided by a particular facility capable of delivering such services. A good example is a medical clinic for neural disorders. However, there may be interaction among facilities that provide specialized services. Thus medical specialists (including neurologists) like to practice in the proximity of a larger hospital, since there is a *complementary* function performed at both facilities. At the same time, hospitals or businesses may compete for a market share of the patients or the customers. Retail stores constitute a prime example, offering *substitutional* or *complementary* goods to lure a customer base.

The way such location-based services are delivered has a direct bearing upon how happy customers are. For example, a vehicle (such as a bookmobile) may make "round robin" deliveries among the customers, or a dedicated vehicle (such as an ambulance) will make an out-and-back delivery, going directly from the hospital to pick up the patient. These two delivery patterns have very different efficiencies and customer expectations, with the former being more efficient and the latter being more satisfactory for both the patient and the attending medical personnel.

The above examples show the disciplinary biases in examining one aspect of a problem vs. another. Thus an urban planner worries about residential location while an industrial engineer worries about factory siting. A clinic worries about both complementary medical facilities as well as substitutional competition among similar clinics, while a shopping mall mostly cares about the competition. On many occasions, the diverse disciplines rarely communicate with one another, even though they are solving related problems. From the analyst's vantage point, however, s/he can see the similarity between the spatial problems they are solving. In this volume [and the companion Chan (2005) volume], we aim to point out these related problems, framing them in the context of common "building blocks." Most importantly, we also try to unify the basic building blocks behind *aspatial* analytics and *spatial* analytics.

II. SPATIAL ANALYTICS

As explained, the complexity of spatial analytics is related to the nature of spatial data, which is multi-dimensional, compared with aspatial data, which is often unidimensional. As explained, the ultimate function of analytics is to extract useful information from data to make the right decisions. It is logical, therefore, to examine the basic process of extracting information and intelligence from a database, whether it be spatial or aspatial. This procedure is often called **data mining**. As applied toward spatial and aspatial data, we present *spatial data-mining* and *attribute-oriented data-mining* techniques respectively (Yeung and Hall 2007). It will be seen that spatial data-mining is a functional extension of conventional data-mining techniques, constructed on the same first principles but using algorithms designed specifically to handle the characteristics and requirement of spatial data. Here are some examples extracted from various parts of this book. As such, it serves as an excellent review of many concepts that we have covered in this volume.

A. Spatial Association

One common data-mining tool is a set of *association rules*. Thus a marketer would like to associate a shopper's preferences to his or her income and educational level, so that the right advertisements can be targeted. When carried over to spatial data, *co-location* is a special type of spatial association. It is defined as the occurrence of two or more spatial objects at the same location or at significantly close proximity to another. Thus a real estate investor might like to know what other buildings will be located next to his or her investment in an apartment building.

Co-location differs from ordinary spatial associations in that there is no natural notion of a transaction between the antecedent and consequent spatial objects. In the above example, there is no implication that there is any interaction between the subject apartment building and the adjacent buildings. By interaction between two building, one may refer to the traffic that goes between these buildings. User-defined neighborhood information is an important factor in constructing co-location rules (Yeung and Hall 2007). In a financial city center, for example, one wants to know how many banks are immediate to one another. We call these adjacent banks *first-order* neighbors. In addition, one may wish to find the neighboring banks that are a bit further away, which are referred to as *second-order* neighbors.

Spatial association goes beyond co-location. In urban planning, one is often interested in how residential development is related to employment. It is conventional wisdom to believe that there is **spatial association** between where one lives and where one works. Such association is manifested in the commuting traffic between the home location and the work location. The home location may not be adjacent to the work location, but it is probably not far enough to exert an inordinate amount of commuting time.

Spatial association information is often sought for each location in a study area. In locating a home, one may ask this logical question: Is this a desirable residential neighborhood? In answering the question, one may consider not only commuting to work, but also accessibility to schools, parks, shopping and entertainment. The concept of spatial association is carried over to regional science. A classic question a regional scientist asks is: how did urban settlements occur historically? In other words, how did Chicago become a trading hub, and developed to be such a big city over time? In this case, trading may range anything from agricultural products to industrial products. Is it the Great Lakes that make such trading possible? Or is it the railroads? Having answered this question, it begs another question: what constitutes the *hinterland* for Chicago, or how should one define the "watershed" area for Chicago? The answer to this question may be different, depending on whether the Great Lakes and/or railroads lead to the development of Chicago. If it is the Great Lakes, then a *study area* (hinterland) may be defined to include the immediate borders of the Great Lakes region. Suffice to say that there are quite a few ways to define a study area depending on the problem context.

Carrying the idea of a hinterland, a home shopper may be interested in school districts or political jurisdictions—beyond other factors. The problem invariably boils down to drawing boundaries on a map, or to divide a region into subregions. We prefer to use the word "subareas" to refer to all these subdivisions, rather than the word subregion. Accordingly, we use the term subareal population and employment rather than (say) sub-regional population and employment. Irrespective of the terminology, robust mathematical models should apply equally well to the different ways to subdivide a region into study areas. A natural way to divide a region into subareas is to use tile-like tessellations such as the **Voronoi diagram**. In this representation, each activity center is defined as the **generator**, for which a "zone of influence" (or a hinterland) is defined, representing the activities which are naturally attracted to this activity center. The activity center may be a place where farmers would bring their produce to market. It has been shown that such a tessellation is consistent with the *Central Place Theory*, which hypothesizes that interregional trade lead toward natural market-place settlements, such as Chicago.

When a spatial unit influences or is being influenced by its neighbors, the subject unit is said to be spatially dependent on its neighbors and vice versa. On the other hand, if activities (such as population and employment) in all the spatial units are truly randomly distributed, they are said to be *independent* of one another. In this case, the assumed value of a spatial unit *i* has no relationship to the value of unit *j*. Thus, one may expect the heights of the residents in zone 1 are usually unrelated to the heights of those living in a neighboring zone 2. Zones 1 and 2 may be related economically, but seldom do tall people or short people chose to live next to one another for company.

Let us carry the concept of association a bit further. Most readers have heard about the idea of a *trend*, which is one form of association. For example, if regional population is observed to be growing over a number of years, it is likely to continue the growth trend. Technically, we say that there is a positive autocorrelation between population over the years. Instead of an association over time, *spatial autocorrelation* measures the correlation of a variable with itself in two locations. Spatial autocorrelation is present when occurrences of similar values cluster together spatially. For example, auto manufacturing activities used to be intense around the industrial belt at the Great Lakes region in the U.S. In other words, there is a relationship among multiple occurrences of values on the same variable (auto manufacturing employment) over a particular region. Given the recent decline in the U.S. auto industry, this relationship might be observed spatially in the past, but may not sustain itself over time.

B. Spatial Clustering

Spatial autocorrelation is used to measure the strength of the relationship among spatial objects of the same type (the *clustering* of auto manufacturing employment in our example). In general, it helps to uncover the extent to which the occurrence of an event or feature at a certain point in space will constrain, or make more probable, the occurrence of another event or feature in its neighborhood. Thus, the intense auto manufacturing activities in lakeside Detroit might have spawned auto-part manufacturing or auto-assembly plants around the Great Lakes region. As an analogy to the Pearson correlation, spatial autocorrelation statistics such as Moran's *I* and Geary's *C* are used to measure this correlation.

To introduce spatial autocorrelation, we discuss in this book the importance of *spatial costs* in organizing the economic activities in a study area. Spatial costs are defined in many different terms. For example, spatial separation is measured in both time and cost. When we wish to convert these diverse measures into a single unit such as *utiles*, we face some challenges. Aside from the conventional "apples vs. oranges" conversion problem, utiles are usually construed as "the more the merrier," while travel time and cost—or *impedance* in general—is exactly the opposite: "small is beautiful." While accessibility to employment is to be maximized, commuting time is to be minimized.

To resolve this dichotomy, we often take an inverse function of impedance (travel time) to convert it from *disutility* (commuting time) to utility (accessibility). This conversion function is sometimes called the *propensity function*. The function can take on the form of a negative power function, $(\text{impedance})^{-b}$, or an exponential function, $\exp[-\beta\,(\text{impedance})]$. Here both b and β are positive calibration coefficients. Irrespective of the form of the propensity function, it is usually calibrated by a *trip-distribution curve*, defined as the frequency with which a trip of certain duration is being executed in the study area. Thus in a city, 50 percent of the commuting trips may be below 25 minutes, 35 percent may be between 25 and 50 minutes, and another 15 percent over 50 minutes. While devised separately by different disciplines, propensity functions and trip-distribution curves have similar shapes, differing by only scaling constants. Both are used to distribute population and employment around subareas in the region.

By now one can see how spatial association leads toward spatial clustering, such as an urban center where population and employment congregate. Most readers are familiar with the concept of a cluster in general, where a cluster is defined as a group of similar objects. For example, one may wish to sort out the random books stored in the basement into fictions, non-fictions, magazines, and textbooks—in this case four clusters. The objective of **spatial clustering** is to find the optimal number of clusters that share common spatial attributes. For example, urban planners have observed that there tend to be high-density developments clusters around subway stations. Many of the algorithms developed for conventional attribute-oriented data mining can be applied or adapted for spatial clustering. For example, traditional partitioning methods such as *k-means* or *k-medoids* are able to capture simple distance relationships and are therefore useful for spatial data mining (Chan 2005). These two methods can be used to discern, for example, urban settlements in the Great Plains, which is a kind of spatial clustering. Similarly, density-based methods, which define clusters as regions of homogeneous characteristics, can be used to detect clusters of arbitrary shapes (Yeung and Hall 2007). Thus, one may wish to discern "wealthy neighborhoods" by virtue of per capita income.

There are also clustering techniques that are especially useful for spatial data-mining applications. These include grid-based methods for raster spatial data and constraint-based methods that allow the inclusion of spatial restrictions on the clustering process. The former was illustrated by image processing applications, where a grid tessellation of gray values is discerned into patterns. The constraint-based method is illustrated by the Benabdallah-and-Wright (B&W) model, as discussed in Chapter 6, in which an area of certain shape or border length is to be discerned. In identifying spatial patterns, whether in land use or satellite images, one can obtain a fair amount of information by observing the "neighborhood" of what one is examining. Thus, a noise pixel can be detected quite clearly as an outlier in an otherwise group of similar gray values in an image. This way, the noise pixel can be removed when *context* is taken into account. On the other hand, districting models have been proposed to divide a community into logical subdivisions as in gerrymandering applications (Benabdallah and Wright 1992; Bennion and O'Neill 1994; Ahituv and Berman 1988). Aside from political districting, the B&W model can group grid cells into clusters to form rectangular-based shapes for image processing.

By these varied examples above, one can see that spatial analytics has its own chemistry that is distinctly different from regular analytics. Yet it is built upon regular analytic tools. Again, this is one of the motivations for composing the present volume. We wish to delineate similarities and differences between regular analytics and spatial analytics. Let us continue this trend of thought below.

C. Facility or Site Location

Two paradigms are used throughout facility location: a geographic representation can be continuous or discrete. The former spreads population and employment via a probability density function over the landscape. The latter distribute them via a probability mass function. One way to compromise such a discrete vs. continuous (or planar) model is through the use of *centroids*. In reality, population and employment distribute continuously over a map. However, there is advantage in modeling the zonal or subareal population or employment as a mass at a single node, which is called the centroid. A centroid is an imaginary node/vertex amid a plane through which activities (e.g., trips) originate from or destine for the subarea. In this case, the centroid is the geographic center or center-of-gravity for the economic activities in this subarea. The use of centroids can also be thought of as a more aggregate representation than its continuous counterpart. The question really boils down to how big one can define a subarea to lighten the computational load, and at the same time, maintain the desired level of accuracy. In this text, centroids and generators are used interchangeably. As the reader recalls, the word generator comes from the concept of a Voronoi diagram, representing the natural gathering places, as discussed above.

There are two traditional criteria in locating facilities. One is the *min-max* criterion and the other is the *min-sum* criterion. For locating a service facility, the farthest demand is to be brought as close to the service facility as possible following the min-max criterion. Applying the min-max criterion results in locating a *center*, whether it be a medical service center or a recreational center. Applying the min-sum criterion, on the other hand, results in a *median*, or a facility that is as close to the demands as possible on the average. Within these two general criteria, quite a few variations are possible, giving rise to a rich array of facility-location models.

We just mentioned that a median is a location that is the closest to the demands on the average. Thus, a retail chain may wish to open a store closest to the population. An **antimedian** is just the opposite. It puts the facility away from the demands. An example is to locate a landfill away from the population for environmental considerations. In short, the median problem minimizes the distance to the total regional demand, while the antimedian problem maximizes the distance to the demand. **Medianoid** refers to a median on a tree (which is a network without any "closed loops" or cycles).

A **medicenter**, also known as **centian** (standing for *cent*er and med*ian*), is a 'hybrid' between a median and a center. It takes care of both the proximity to demands as well as the reduction of the most adverse exposure. One can argue this is the best criterion for locating an incinerator—close in general to rid of garbage, yet not too close for environmental considerations. **Anti-medicenter** is just the opposite of medicenter. It maximizes the sum of the weighted distance where the demands serve as weights. Yet at the same time, we minimize the maximum weighted distance. It may be the best for locating an airport, which should be a reasonable distance away from the regional population, yet within reach for the most remote residents.

In discrete facility-location models, a **condorcet point** is any point in the network that is closest to *most* of the demands. A **Simpson point**, on the other hand, is illustrated by an example by Bhadury and Tovey (2010). Two competing firms vie for a common market by locating their own facilities to sell an identical product. Of the two competing firms, one is designated as the leader that decides to locate one of its own facilities first. The leader is aware of the fact that after it has entered the market, the rival firm, denoted as the follower, will locate one of its own facilities in such a manner as to take away as much a market share from the leader as possible. Given this, the decision problem facing the leader is to find an optimal location, the Simpson Point, such that the maximum market share that is lost to the follower is as small as possible. Both Condorcet and Simpson points are relative, rather than absolute, concepts.

A well-known fact in linear programming is that the optimal solution has to occur at an extreme point of the feasible region. This property is carried over to network facility-location models. For example, the optimal siting is often found at a node/vertex. Thus a fire station is to be sited at an intersection of a street network. This is not an intuitive result by any means, since there is no reason a priori why the optimal location cannot be on an arc or at any other place. This *nodal-optimality* property, where identified, does allow us to design some computationally-efficient solution-algorithms. Available evidence suggests that certain extremal conditions also exist in planar location models, in which the facility can be sited theoretically at any point in the Euclidean space. For example, the optimal airport between three cities is often located at one of the cities. In other words, the optimal site is at a vertex of the triangle formed by the three cities as vertices, rather than somewhere inside the triangle.

D. Routing

As discussed above, a location is picked considering accessibility to economic, social, and recreational opportunities. To reach these opportunities, trips may have to be executed either by the client population or the provider. In this case, the population follows a route to the goods and services, or the goods or services have to be delivered by the provider to the population. When a special delivery is made by the provider, the vehicle used for the delivery may be productive only in one direction, namely on the way to deliver the goods. The return trip is often empty and not productive, unless another load is backhauled to the provider. On the other hand, if an individual combines several 'errands' on a trip, s/he completes a "round robin" visit to several service providers one after another. We call this a *tour*, which can likewise be executed by the provider to deliver the goods or services. Node routing is to assign and sequence discrete stops, and arc routing is to assign and sequence street segments. Arc routing is more specialized and occurs when vehicles visit every (or most) address on block segments, as in meter reading, mail delivery and garbage pickups.

The algorithms underlying routing products typically involve a combination of *integer programming* methods and heuristics. Other algorithms are based on heuristics, artificial intelligence and expert system approaches, rather than traditional mathematical programming. For example, a space-filling curve transforms a two-dimensional map into a single dimension. By observing the clusters in the single-dimension line instead of proximity in two dimensions, vehicle tours can be constructed much more conveniently for each cluster of demand points. (Please see an example in Section 3.3 of Chan (2005)).

In Chapter 3, the *hypercube* model dispatches a fleet of service vehicles in response to calls. A vehicle at a depot is either free or busy, as represented by the binary 0-1 variable. For two depots with a vehicle at each, (0, 0) denotes both vehicles are free and available for service; (0, 1) means only the vehicle from the first depot is free; (1, 0) means only the vehicle at the second depot is available; and (1, 1) says both are busy. The four states of the system—(0, 0), (0, 1), (1, 0), and (1, 1)—can be plotted as four nodes/vertices in a graph. Arcs between the nodes describe the possible transitions between these states. Such a state-transition graph resembles a rectangle, characterized by the four nodes/vertices and arcs representing the possible transitions between the states. When there are three depots, the graph resembles a cube. In the general case when there are any number of depots, the graph is a 'hypercube,' and hence the name hypercube model. Technically speaking, it is a spatial queuing-model that caters for random calls or demands that arrive at a fixed arrival rate. It is a location-routing model that dispatches vehicles from the closest depot when a vehicle is available.

III. SOFTWARE

Let us now assemble some information technology (IT) tools to support the above analytics, whether it be general analytics or spatial analytics. If one adopts the broader definition of analytics to mean analyses to support decisions, there would literally be no limit to the number of software available. One would need to go from basic mathematical software to modeling software. It is not our intention, nor are we prepared, to cover this broad subject. We have to limit ourselves to software that directly support the subjects covered by this book.

To facilitate analysis, many books on analytics package software with the book. We decided to take a different approach. In view of the huge number of both commercial and public-domain software and the pace they are changing, we decided to provide an objective evaluation of existing and emerging software instead, guiding the reader to make his/her own decision on the software most suitable for him/her application. The reader can then go to the vendors. Many of them offer trial versions of their software, with which the user can check out the most current software in detail. Alternatively, the reader can seek "freeware," which are increasingly available. Here, we like to guide the readers in two ways. In the present chapter, we provide a top-down analysis of software, spelling out the salient features that a user is advised to look for. This will include both commercial software and "freeware." In Chapter 8, we document in more detail selected software packages, including *open source* software, which make available the source code for further development.

The software discussion will proceed in two steps. We start with the general-purpose analytics software, and progress toward those that support spatial analytics directly. As it turns out, a general purpose software is by-and-large commercial software, since the bottom line would dictate that the vendors develop software that has a wide audience. On the other hand, research and educational institutions have more luxury to look out on the horizon, venturing into developmental software that cater for specialized applications such as spatial analytics. In the following sections, the general features of a software package is outlined, together with guidelines for selecting an appropriate software. A judgmental screening and details of the screened software are documented in Chapter 8, entitled "A Software Survey of Analytics and Spatial Information Technology." To the extent that developmental spatial information technology, including educational software, reflect the trend the field of is heading, they are mostly discussed in the current chapter. Judgmental screening is performed for more general analytics software in Chapter 8, a majority of which is commercial software. Chapter 8 is therefore mainly intended for those who are considering acquiring software for day-to-day use.

A. Commercial/Licensed Software

Software for both regular analytics and spatial analytics will be discussed in terms of commercial software and public-domain (free) software respectively. We break down commercial software in several categories, including simulation software, statistical software, optimization software, decision-analytic software, Geographic information system (GIS), and image precessing software. The sequence parallels the order in which these techniques were discussed in the book. Chapter 8 goes further by introducing some guidelines for **spreadsheet modeling**, which is increasingly popular. We also include a survey of software for general **mathematical modeling**, such as MATLAB and MATHEMATICA. In the absence of commercial packages, we review the prevalent public-domain software for spatial analytics. Those include educational software. We end up the review with the specialized software we developed for this book. In short, we start with a broad base and build the "pyramid" one layer at a time. The strong broad base of general analytics software is in the foundation, with the tip of the pyramid represents more fragile research and development efforts.

1. Regular Analytics Software. Starting with regular analytics software, the following sections provide more than just a set of criteria for choosing a software package. The criteria or salient features serve as a review of the state-of-the-art in both regular and spatial analytics tools, whether we are talking about statistics, simulation, optimization, decision analysis, GIS, image processing, facility location, or routing.

Statistical Software

At the foundation of analytics is statistical analysis, which provides the inference from and fidelity of the measurements taken from samples. Similar to other software, statistical software transitioned from mainframe computers to desktop personal computers in the early 1980's. In a survey, Woodward and Elliott (1983) found out that while many computers were supported, there was a wide variation in the price of the software and in its capabilities. Most of the packages do not include box plots, confidence intervals, survival analysis, categorical-data analyses other than chi-square contingency-table analysis, multiple comparisons, or nonlinear regression. Many will not read a rectangular data file created externally to the program. Woodward and Elliott's experience with several of the packages had revealed some glaring calculation errors in the coding.

Three decades later, desktop statistical packages have seen dramatic improvement. The statistical software available in the market today range from general tools that cover the standard techniques of inference and estimation to specialized activities such as nonlinear regression, forecasting and design of experiments (Swain 2007a). As with other analytic software, products that provide statistical add-ins to spreadsheets remain common. The spreadsheet is the primary computational tool in a wide variety of business settings, familiar and accessible to all. Many procedures of data summarization, estimation, inference, basic graphics and even regression modeling can be added to spreadsheets.

By and large, dedicated general and special-purpose statistical software has a wider variety and greater depth of analysis than add-in software. For many specialized techniques such as forecasting, design of experiments, special-purpose statistical packages are appropriate. Moreover, new procedures are likely to become available first in these (specialized) statistical software and only later be developed into the add-in software. In general, these statistical software plays a distinct role on the analyst's desktop. Assuming data can be freely exchanged among statistical-software applications, each part of an analysis can be made with the most appropriate (or convenient) software tool.

An important feature of statistical programs was the importation of data from as many sources as possible. This will eliminate the need for data entry when data is already available from another source. Most programs have the ability to read from spreadsheets and selected data-storage formats. Also highly visible is the growth of data warehousing and "data mining" capabilities, programs and training. Data-mining tools analyze data from a variety of sources to look for relations that would not be possible from the individual datasets.

Graphics provide a powerful way of presenting data, and the dynamic exploration of various graphics can be a powerful method of uncovering underlying relations within the data (Swain 2009). Since plotting is limited to two- and three-dimensional projections of data, several methods have evolved to form a coherent view of the complete picture. In the matrix plot, for instance, variables are listed by row and column and the pairwise scatter plots occur in each position (sometimes with the univariate histograms in the diagonal positions). Hence, the scatter plots in each row represent the joint distribution of that variable with the variables in the respective columns.

Of course, these marginal views cannot provide the entire story. For example, correlations may exist within a group of variables that are not observable by the pairwise scatter plots. In these cases, additional insight can be obtained through transformation of coordinates based upon principal components or factor analysis. Such techniques are often used when the underlying variation consists of groups of related variables, and these underlying factors are often fewer in number than the number of variables that are directly observed.

Interactive graphical methods can also be used to explore relations within data. "Brushing" is a method in which a point or a group of points on a given plot can be highlighted in linked displays or simply used to provide access to their location within the data for detailed examination. Another form of interactive graphics is obtained through 'slicing.' In this case a variable is designated for slicing, and the data is divided into sets above and below the slice value of the variable. As the slice point is varied one is able to highlight the characteristics of the linked displays. The value of the set of residuals can reveal that the positive residuals come from a limited set of data, such as a particular industrial sector. Similar inferences can be made for negative residuals.

Forecasting software can fall into one of three categories according to Yurkiewicz (2010). Automatic forecasting software will quickly do an analysis of the data and then make the forecasts using a methodology that it deemed the most appropriate. The chosen technique may come from the software, minimizing some statistics such as the Akaike Information Criterion (AIC), the Bayesian Information Criterion (BIC), RMSE, etc. The software will give the calibrated parameters of the model, confidence intervals for the forecasts, plots and various statistical summaries. The user always has the option of bypassing the chosen methodology and specifying some other technique. Semiautomatic forecasting software asks the user to specify a methodology from a list of available techniques. Finally, manual software requires the user to specify both the technique and the parameters for the model. In an application, the user must decide whether to use a dedicated forecasting program or a general statistical product that has the desired forecasting capabilities. Dedicated products are more likely to be automatic programs. They may also offer more sophisticated forecasting techniques than general statistical programs.

When looking for a time-series software, it is advisable to see whether it provides a fully automatic expert system for univariate and multivariate time series, following the Box-Jenkins modeling framework (Flores-Cerrillo 2010). Equally desirable is to see whether it incorporates intervention detection. Top performing packages will automatically identify the best subset of seasonal autoregressive-integrated-moving-average (ARIMA) models for a given problem. For example, dedicated forecasting software often cover ARIMA intervention, multivariate ARIMA transfer functions, etc. They may even have automatic outlier-detection capabilities.

In summary, here are some factors for consideration when selecting a general statistical software (Swain 2009). They are compiled as a checklist in Table 7.1.

Simulation Software

Assisted by statistical inference, simulation is a very robust tool that tackles a wide variety of problems. We will concentrate on products that run on personal computers to perform *discrete-event* simulations. Hlupic (1999) discusses the users' requirements of simulation software. Their surveys, conducted in 1990's, show that simulation software used by participants in these surveys is predominantly easy to use, with good visual facilities, but too limited for complex and nonstandard problems. It is also too expensive, and at the same time incapable of providing adequate guidance in experimentation. Areas for improvement dominantly refer to more assistance in experimental design. There is a need for easier-to-learn-and-use packages and improved software compatibility. The survey, conducted twice, once in 1992 and another time in 1997, indicate the trend of using several simulation packages

Table 7.1 FACTORS IN SELECTING A STATISTICAL SOFTWARE

□ Operating systems: Consider the suggested minimum random access memory (RAM) for PC/Windows (XP, Vista), PC/LINUX, UNIX, and others.

□ What are the import formats and export formats?

□ Are the following tools available?

- Online help/tutorials,
- Exploratory data analysis,
- Data edit/transformation,
- Graphics, dynamic graphics (plot linking, data brushing),
- Tabular outputs,
- Descriptive statistics,
- Distribution fitting,
- Non-parametric statistics,
- Power/sample size computation,
- Quality and process capability,
- Six-sigma analysis,
- Analysis of variance/generalized linear model,
- Design of experiments,
- Response surface methods,
- Regression fitting,
- Regression selection,
- Logistic regression,
- Nonlinear regression,
- Time series/forecasting,
- Multivariate statistics,
- Clustering/classification,
- Reliability/survival analysis,
- Data/database management,
- Data mining,
- Classification and regression trees,
- Re-sampling/Monte Carlo.

□ Pricing Information for these options: Commercial—Single machine and site? Educational—Single machine and site? Demo/Student—Single machine and site?

rather than only one package. The apparent reason is that a single package probably is not adequate for all applications.

The range and variety of simulation products continue to grow since the now dated surveys in the 1990s, reflecting the robustness of the products and the increasing sophistication of the users (Swain 2009). Software ranges from specialized applications such as health care and logistics to general purpose simulation. They include information about experimental run control (e.g., batch run or experimental design capabilities) and special viewing features, such as confidence intervals and risk measures. Many packages can produce animations or demonstrations that run independent of the simulation software itself. Of equal importance is the classic input PDF or probability mass function required of any discrete-event simulation. These input distributions can be generated from resident programs or linked programs from an independent source such as the Stat::Fit software.

Newer simulation products are beginning to provide autonomous agents that can act on their own and interact among themselves (Swain 2007b). These **agent-based simulations** have evolved from earlier studies of complex systems whose behavior has defied easy explanation. Models of biological systems involving interacting members have been built, for example, to understand how these complex behaviors have led to social behaviors in foraging. Generalizing what has been learned from these studies has led to simulation agents with the ability to sense their surroundings, interact with other agents, reason and choose a course of action.

It has been a continuing trend to include optimization features as part of simulation. This is accomplished either by a built-in optimizer or by a configurable module such as OptQuest (as developed by OpTek Systems Inc.). A majority of the more "substantive" software has some kind of optimization capability, and the number is increasing. Care is also taken by many software vendors to make the simulation output compatible with common office suites such as MS Office. Output statistics can typically be exported in MS Excel or Access formats for further analysis.

Commercial simulation products in entertainment (such as video games) have created lucrative markets that have spurred an increasingly rapid pace of innovations. Up to this point, training simulators have been rather specialized and expensive. It is largely the domain of large commercial enterprises. As it becomes easier to build realistic **virtual reality** simulations, these can now be used in the classroom to make more realistic (and compelling) case studies. It provides some of the background that only industrial experience could provide before.

When shopping for a software, here are some factors for consideration (Swain 2009). They are compiled as a checklist in Table 7.2.

Optimization Software

Under the optimization label, a majority of software has the format of minimizing or maximizing linear functions subject to linear equalities and inequalities in numerical decision variables (Fourer 2009). All products provide for continuous variables that may take any values between their bounds, and many also accommodate integer variables that are limited to whole-number values. The continuous and discrete problems that are described by these variables are the linear programs (LPs) and integer programs (IPs) respectively. In between LPs and IPs is mixed-integer programs, or MIPs. Some of the products handle other kinds of discrete variables (such as binary variables) and constraints, as well as varied non-linearities. Indeed a trend toward greater generality in recent years continues to be seen. Combinations of MIP and non-linear programming have been a recent focus of intensive solver development.

Solver software takes an instance of a model as input, applies one or more solution methods and returns the results. **Modeling software**, on the other hand, does not incorporate solution methods. It is typically designed around a computer modeling language for expressing a mathematical model and offers features for reporting, model management and application development. Many have a translator for the language on top. Numerous solver and modeling products have been developed as independent applications. Thus, solvers typically support links to several modeling systems, and modeling systems offer links to several solvers. In some cases the two may be acquired as separate products and linked by the purchaser, but more commonly they are bought in bundles of various kinds. Most modeling-system developers arrange to offer a variety of bundled solvers, providing modelers with an easy way to benchmark competing solvers before committing to purchase one. Some solver developers also offer bundles with modeling systems. A number of the latter developers also offer integrated systems that provide a modeling environment specifically for their own solvers.

Table 7.2 FACTORS IN SELECTING A SIMULATION SOFTWARE

□ Typical applications of the software

□ System requirements such as RAM and operating systems

□ Model building features:

- Graphical model construction ability (utilizing icon or drag-and-drop),
- Model building via user programming or through access to programmed modules,
- Runtime debug capabilities,
- Types of input-distribution fitting,
- Types of output-analysis support,
- Batch run or experimental design,
- Distribution fitting,
- Optimization capabilities,
- Provision for code reuse through objects, templates etc.,
- Can a completed model be shared with others who might lack the software to develop their own model?
- Mixed discrete/continuous modeling capabilities (particularly the levels, the flows, etc.)?

□ Animation:

- Animation capability,
- Real-time viewing?
- For presentation purposes, can the MPEG[1] version (with compressed audio and visual digital data) be run independent of the simulation?
- What are some compatible animation software?
- Three-dimensional animation capability?
- Can the software import CAD drawings?

□ Support/Training: User support/hotline available? User group or discussion area exists? Are there training courses? How about on-site training? Is consulting available?

□ Price: For standard vs. student version.

Since optimization models are usually developed in the context of some larger algorithmic scheme or application (or both), the ability of the software to be embedded in an application is often a key consideration. Thus, although virtually any of the listed products in Chapter 8 can be run as in independent application in a stand-alone mode, many are available in **callable library** form, often accessible as class libraries in an object-oriented framework. Solver systems have long been available in these ways, with an application-specific calling program taking the place of a general-purpose modeling environment. Modeling systems have increasingly also become available for embedding, so that the considerable advantages of developing and maintaining a modeling language formulation can be carried over into application software that solves instances of a model. It is possible to embed an entire modeling system, or a particular model, or an instance of a model. Not all systems provide all possibilities, so some study is necessary to determine which products are right for a given project.

The application development environments provided by spreadsheet and database programs have proved to be particularly attractive for embedding optimization software. At the least, most modeling environments can read and write common spreadsheet and database file formats. Spreadsheet packages can also accept solver add-ins, whose appeal to users and convenience for development are widely appreciated. The solver add-ins that come packaged with spreadsheet products are effective only for small and easy problems, but independent developers offer much more powerful spreadsheet options in recent years. Some can work with a variety of spreadsheet functions that go beyond the smooth arithmetic functions assumed by classical optimization software. Setting aside spreadsheets, several scientific and statistical packages also offer LP software add-ins specifically for their products. MATLAB appears to be the most popular in this respect. Another example is SAS.

Virtually all modeling systems and solvers can also handle model instances expressed in simple text formats. These include the "MPS" format dating back many decades and various "LP" formats that resemble textbook examples complete with $+$, $-$ and $=$ signs. These formats mainly serve to submit bug reports and for communicating benchmark problems. Modeling systems use much more general and efficient formats for communicating problem instances to solvers and for retrieving results. Each uses its own format, unfortunately, so that every modeler-solver link requires a different translation. There is continuing interest in a superior standard form that could express problem instances of various kinds, in ways that would help to integrate optimization software with Web communication standards like XML. Progress has been gradual, however, and no definitive standard form has been adopted as yet.

Solution methods have continued to be refined for speed and reliability. For LPs a choice between primal simplex, dual simplex and interior-point methods is standard. The "bag of tricks" that make up the typical MIP branch-and-bound solver continues to grow even after decades of attention, with increasingly sophisticated features such as branch-and-cut, branch-and-price and feasibility-seeking heuristics becoming available to a broader range of users. These refinements make more integer programs tractable but also place more responsibility on the user to study and select wisely among available options. Although MIP solvers attempt to choose options according to characteristics of the problem at hand, these default choices cannot be relied upon to work well for all hard MIPs. Users may find it necessary to "tune" algorithmic options through experimentation; some solvers provide suggestions for making good choices, but explicitly automated tuning is still at an early stage.

Many packages seek to address their users' needs by supporting varied specializations and generalizations of LPs and MIPs. In the area of discrete optimization, the ideas underlying branch-and-bound searches are sufficiently powerful to handle broader classes of constraint types. **Special Ordered Sets** (SOS) exploit special structures in MIP models during the solution phase. Indeed, MIP solvers have long accommodated variables that take values from an arbitrary list (via special ordered sets of type 1 or SOS1 search rules) and objectives or constraints that incorporate non-convex piecewise-linear terms (via SOS2 rules). Here the SOS1 (or SI in brief) rule suggests that at most one variable within a set can have a non-zero value. They most frequently apply where a set of variables are actually 0-1 variables: in other words, we have to choose one from a set of possibilities. For example, only one out of *N* options can be selected.

The SOS2 (or S2 in brief) rule suggests that at most two variables within a set can have non-zero values. And if two are non-zero these must be consecutive in their ordering, or the two non-zero values have to be adjacent. They are the natural extension of the concepts of Separable Programming.[2] Separable programming basically replaces all separable functions, in objectives and constraints, by piecewise linear functions. When embedded in a Branch-and-Bound code, SOS2 enables truly global optima to be found, and not just local optima. For example, a non-convex separable function can be linearized using SOS2.

Aside from LP, IP and MIP, many packages have solution capabilities for stochastic programming and nonlinear programming in general. Stochastic programming is receiving increasing attention, echoing the preference for embedding optimization routines within discrete-event simulation software, as discussed in the section under Simulation. A major complication of nonlinear programming, however, is that it requires computing derivatives (Nash 1998). Modeling languages provide assistance here, either by computing approximations to the required derivatives, or by deriving and programming the derivative formulas as the model is processed. If a modeling language is not being used, the tedium of derivative calculations can be alleviated through the use of software for "automatic differentiation" (such as ADIFOR). Such software analyzes the formulas for the nonlinear functions and generates software that will evaluate the derivatives. This is not the same as "symbolic differentiation" — the technique used in packages such as MATHEMATICA. Unlike symbolic differentiation, automatic differentiation can be applied even in cases where the nonlinear functions are only defined in terms of other software. The model functions need not be expressed in "closed form." These two resources (modeling languages and automatic differentiation) remove much of the tedium associated with specifying a nonlinear model. They can greatly simplify the task of preparing a model for the optimization software.

Conic optimization problems are a class of convex nonlinear optimization problems, lying between LP problems and general convex nonlinear problems. Among others, convex quadratic programming and quadratically constrained programming problems can be formulated as **conic optimization** problems. A conic optimization problem can be written as an LP—with a linear objective and linear constraints—plus one or more cone constraints. A cone constraint specifies that the vector formed by a set of decision variables is constrained to lie within a closed convex pointed cone. In linear algebra, a convex cone is a subset of a vector space that is closed under linear combinations with positive coefficients.[3] If the origin belongs to a cone, then the cone is said to be pointed. Otherwise, the cone is blunt.

LP further extends to semi-definite programming (SDP). SDP is a subfield of convex optimization concerned with the optimization of a linear objective function over the intersection of the cone of positive semi-definite matrices with an affine space. In linear algebra, a positive-definite matrix is a matrix which in many ways is analogous to a positive real number.[4] An affine space is a vector space that has forgotten its origin. Imagine a linear combinations of two vectors in which the sum of the coefficients is 1. Such an affine structure describes the same point with the same linear combination in their respective frames of reference, wherever the origin may be. An underlying set with an affine structure is an affine space. Once again, an affine space is what is left of a vector space after one has forgotten which point is the origin.

Problems of these types find varied applications in engineering and design, and provide strong approximations to some hard combinatorial problems. A search of the Web readily yields several collections of test problems. Interior-point methods extend to solve these problems, though not so easily as in the case of LPs. Problems of these kinds are becoming more familiar as modeling languages and problem formats catch up with them.

The range of supported platforms continues to be stable (Fourer 2009). Windows remains universal, and Linux has become nearly so for products other than spreadsheet add-ins. Among other Unix variants, Solaris, HP-UX and AIX are still quite common. Support for Apple computers has increased substantially, though primarily through ports to the Unix shell of MacOS System X, rather than through the creation of new versions that conform to a more standard Macintosh look and feel.

Multiprocessor versions for shared memory have become widely available, as multi-core processor architectures have become the standard and two quad-core processors have become a readily obtained configuration on high-end PCs. Support for distributed memory remains relatively rare, despite continued general interest in "grid computing" and networks of workstations. Distributed processing seems a natural fit for branch-and-bound methods in integer programming, which solve independent subproblems at nodes of a huge search tree. However, promising experiments with this approach do not seem to have led yet to much commercial support.

Over the past years, we have witnessed the steady improvement of general-purpose solvers. According to Yunes, Aron and Hooker (2010), the next software development is to combine mixed-integer linear programming, constraint programming,[5] and global optimization[6] in a single system. Recent research in the area of integrated problem solving suggests that the right combination of different technologies can simplify modeling and speed up computation substantially. Many traditional optimization techniques can be seen as special cases of a more general method, one that iterates a three-step procedure: (1) solving relaxations, (2) performing logical inferences, and (3) intelligently enumerating problem restrictions. A major advantage of an integrated solver is precisely that it can exploit structure while remaining a general-purpose solver and providing the convenience of current commercial systems. Future development of related systems will presuppose less knowledge on the part of the average user to solve less difficult problems. At the same time, it will give experts the power to solve harder problems within the same modeling framework. One way is to increase the library of meta-constraints, solver types, constraint relaxations, and search strategies, with the goal of accommodating the full spectrum of problems to be solved.

When shopping for a software, here are some factors for consideration (Fourer 2009, Nash 1998). They are compiled as a checklist in Table 7.3.

Decision Analysis Software

The information provided here is intended to help analysts select a tool set that fits the specific problem they face or maybe even a general-purpose package for the long run (Buckshaw 2010; Maxwell 2008). While this is the motto for this entire chapter, we wish to highlight it for decision analysis software. When shopping for decision analysis software, it is particularly advisable to focus on the potential tool's ability to fit the specific problem or class of problems. The potential user is

Table 7.3 FACTORS IN SELECTING AN OPTIMIZATION SOFTWARE

- □ Software description:
  - Type: Solver, modeling environment, integrated solver and modeling environment;
  - Forms: Independent application, procedure/callable library, object/class library, source code, add-in to MS Excel.
- □ Platforms supported include 32-bit and 64-bit configurations on: PC/Windows, PC/Linux, other Unix-based, other OS
- □ Microprocessor support can be either shared memory or distributed memory
- □ Size of Problems solvable by this system:
  - Is the largest size limited by internal restrictions, maximum number of constraints, available memory, available disk space, or processor architecture?
- □ Demo/student version:
  - What is the maximum allowable number of constraints, variables, integer variables, non-zeroes?
- □ Is it a free or open-source software?
- □ Access to the NEOS server?
  - NEOS provides an XML-RPC server that communicates with clients for submitting and retrieving jobs. Users only need a definition of the optimization problem; all additional information required by the optimization solver is determined automatically.
- □ Pricing information for commercial, educational, and demo/student versions on a
  - Single machine,
  - Floating licenses where available,
  - Site license where available.
- □ Data compatibility:
  - Capacity to read spreadsheets, write spreadsheets, read database, write databases, read and write text?
- □ Solvers or modeling environments:
  - Are the solvers/modeling environments that link to the product bundled as single package, or available separately?
- □ What are the model formulations that are supported?Variable Types:
  - Integer, binary; semi-continuous;
  - Arbitrary discrete (special ordered sets SOS1);
  - Piecewise linear (special ordered sets SOS2);
- □ Other Constraint and Objective Types:
  - Convex quadratic objective, 2nd-order cone, general convex, general nonlinear, other.
- □ Available Algorithms:
  - Linear programming: primal simplex, dual simplex, interior point?
  - Integer programming: branch-and-cut, branch-and-price, heuristics for seeking feasible solutions?
  - Other algorithms, such as derivative calculation requirements for nonlinear programs and the corresponding derivative calculation tools? How about stochastic programs?
- □ Available Utilities:
  - The presolve pre-processor to simplify an optimization model before solving it; infeasibility diagnosis; other?

advised to carefully evaluate the software in relation to the situational factors that are relevant. If the goal is to add a package or two to the general toolkit, then a package or combination of packages that provide balanced support across the spectrum of situations and the entire decision-analysis process is the best investment. If the problem calls for involving multiple stakeholders and multiple competing attributes, then tools that emphasize group support and value elicitation are worth exploring. Problems involving large uncertainties, diagnosis, complex interdependencies or risk analysis would benefit most from tools such as influence diagrams, Bayesian networks or one of the Monte Carlo modeling tools.

Due diligence in selecting a decision-analysis package should include thinking about the following critical questions (Maxwell 2008):

1. Are there a single stakeholder or multiple decision-makers?
2. Will the stakeholders participate in the decision conference, or will they be periodically presented results only?
3. Will there be a choice of a single alternative or a portfolio of alternatives?
4. Do stakeholders have multiple, conflicting objectives that must be reconciled?
5. Is there significant uncertainty in the decision outcomes?
6. Is it a single, one-time decision or a sequence of decisions over time?

Most hard decision situations require decision-maker(s) to make trades among a complicated set of competing objectives. There is a number of multicriteria decision-making techniques implemented in the available software. Multi-Attribute Utility Theory (MAUT) and the Analytic Hierarchy Process (AHP) are the most prevalent. Most of the packages indicate that they implement MAUT. In addition to these approaches, ordinal ranking techniques are available in some of the software packages and can be quickly implemented to develop a first-order set of weights for a decision model. The quick technique just might be good enough to meet some of the analysis goals.

More important than the software, it is critical that analysts understand that different techniques have different underlying axioms and different philosophies about how decision models should be formulated. The best example is MAUT vs. AHP, two very different techniques with different theoretical underpinnings. Various approaches have strengths, weaknesses and limitations that deserve some research before they are applied. Whichever technique is applied, it is important that analysts ensure that both the relative importance of attributes and the range within which each attribute varies are clearly presented to the stakeholder for consideration as an integral part of the elicitation process. Considering only importance increases the risk that the model will produce unreliable results. This point has been discussed in Chapter 5.

Uncertainty is also almost always a factor when making hard decisions. How it is addressed varies among the packages. How it is best addressed depends on the nature of the uncertainty, how the model is being developed, the data that is available and the resources that are available for model development. Tools are available for eliciting probability judgments from experts. Often, these judgments are placed in a decision analytic model called an *influence diagram*. These diagrams are designed to combine an intuitive, visual presentation of the relationships among the variables with a sound underlying mathematical representation of their joint probability distribution.

Let us go back to the early days of influence diagrams and other decision analytic models that considered uncertainty explicitly. In that era, solution time and computer memory for models were very important considerations. As an example, one influence diagram model developed in the very early 1990s possesses almost two million solution paths. It took approximately two hours to solve. Today, the same model—using newer versions of the influence diagram software and a current notebook computer—solves in less than three minutes. This power allows us to represent and solve increasingly complicated problems. It also allows analysts to exercise the models we develop more rigorously.

In some cases the analytic team might have large quantities of data that can be used to formulate the probability model. Some of these packages (as well as some statistical packages) have learning algorithms that will build the joint probability distribution from the available data. If this option is available, the analysis team should be certain to supplement the automated effort and involve subject matter experts in the review of the resulting model. The experts can help find errors in the data and, just as importantly, they can supplement the model with knowledge they possess. Combining what is learned from data and what is learned from experts usually yields a better model and results in a higher likelihood that the effort will be successful.

A final consideration should focus on whether the model is addressing a single decision in time or a sequence of decisions over time. Virtually all of the packages will consider a single decision. The influence diagram packages and some of the Monte Carlo packages will also consider multiple decisions that might unfold over time. In influence diagrams, this situation is represented as a sequence of decisions, likely with uncertainties that will resolve over time spaced in between decisions. Sometimes a hard decision actually consists of a set of smaller decisions that either occur over time or can be thought of as a package. A technique for representing this type of situation is to use a sequence of decisions or an alternative-generation table. This technic is virtually supported by all of the software packages.

Looking toward the future, here is an observation according to Buede (undated). Future improvements that will be least likely to make substantial advances involve embedding the analyst's wisdom and knowledge in the software for the users who are less-skilled analysts. These improvements should include problem structuring and elicitation features. Buede believed these capabilities might exist in the distant future, but are not very likely in the near future. Important as it may be, market for this software will always be a small, specialty market until the less skilled analysts are better prepared. Buckshaw (2010) echoes this sentiment by suggesting that software vendors should embed some form of coaching into their products so that even a novice can be confident that their models are producing sensible results.

When shopping for a software, here are some factors for consideration (Maxwell 2008). They are compiled as a checklist in Table 7.4.

2. Spatial Analytics Software. As mentioned, there is an increasing interest in analytics that support spatial analysis. Here we discuss the salient features of spatial analytic software, including GIS, image processing, and vehicle routing. Karimi (2009) has taken this idea a step further and proposed the term geoinformatics. In his terms, geoinformatics is the science and technology of gathering, analyzing, interpreting, distributing, and using geospatial information. It includes the topics of spatial databases, mapping and visualization, analysis, ontologies, distributed geoprocessing, location-based services, and management.

Table 7.4 FACTORS IN SELECTING A DECISION ANALYTIC SOFTWARE

- □ Operating Systems:
    - Is the user's operating system Windows, Mac OS, Unix, or other?
- □ Applications:
    - What is the best software if multiple objectives are considered?
    - How does the software represent and analyzes uncertainty for a probabilistic application?
    - How does it represent and analyze probabilistic dependencies?
    - Where applicable, how does it model sequential decision making, portfolio decision making, and/or multiple-stakeholder collaboration?
- □ Software Features:
    - Can the software import a database or spreadsheet?
    - Does it export presentation graphics?
    - Does it interface with the EXtensible Markup Language (XML) that facilitates transport and storage of data?
    - Does it accept Application Program Interface (API), such as embedding a decision support system?
    - Can model segments be copied or moved easily?
    - Can model structure be displayed on screen or printed?
    - Can a user protect his/her data from other users?
    - Does the software support explicitly group elicitation? How?
    - Does it support simultaneous viewing?
    - Is a record of model evolution kept?
- □ Does the software support: Multi-objective decision analysis? Multi-attribute utility theory? Analytic hierarchy process? Or other algorithms?
- □ Is pricing information available for commercial, education, enhanced/high performance licenses?
- □ Are there size limitations in the following: Number of alternatives? Number of levels in value or decision tree? Number of states of a node in a tree?
- □ Is graphical elicitation techniques available for the following:
    - Model structure/brainstorming?
    - Value functions/scores?
    - Value weights, probabilities, risk preference?
    - Can probabilities or weights be defined as variables that can be operated on?
    - Are graphical sensitivity analyses possible on either weights or probabilities?
    - Can analytical results be portrayed graphically?
    - Can the user document structure or judgments with text?

GIS Software

While its functionality is much broader, GIS is often thought of as computer systems for managing a spatial data structure (Lee and Zhang 1989). In selecting a proper GIS software, therefore, one needs to consider both hardware and software configurations that handle large map and image databases, often in a distributed computer network. The main hardware factors that influence the performance and capacity of a GIS, perhaps more than other computer systems, are word length, main memory size, processing speed, size of external storage, and data transfer rate between external and main memories. Current GIS software packages include management systems, logic programming, object-oriented programming, and object-oriented databases. The complexity lies in the need to integrate geometric and non-geometric data and the need for a distributed system. Given its complexity, the cost of GIS software development is generally high, and this reflects in its market price.

There are five essential elements in a GIS according to Star and Estes (1990): **data acquisition, preprocessing, data management, manipulation and analysis**, and **product generation**. These elements need to be properly considered in acquiring a GIS. We have addressed data acquisition already in Chapter 6. The remaining four elements are intimately related to the way hardware and software are configured. Preprocessing involves manipulating the data in several ways so that they may be entered into the GIS. Two of the principal tasks of preprocessing include data format conversion and identifying the locations of objects in the original data in a systematic way. The first involves converting paper maps and transparent overlays to computerized data sets. Modern-day scanners and digitizers greatly assist in the process, but much of the work, as is usually the case, still rests with the human. The second task is to determine the characteristics of any specified location in constructing the data layers in the GIS system. It is clearly a labor-intensive and skill-intensive effort to ensure that the resulting database can be of maximum value to the user.

Data management functions govern the creation of, and access to, the database itself. These functions provide consistent methods for data entry, update, deletion, and retrieval. Modern database management systems isolate the users from the technical details of data storage, such as the particular data organization on a mass storage medium. When the operations of data management are executed well, the users usually do not notice, nor do they care, about the intricacies of the information processing technology. When they are done poorly, however, everyone notices the slowness of the system, the cumber with which the system operates, and the frequent disruption. Finally, data management concerns include issues of security. Procedures must be in place to provide different users with different kinds of access to the system and its database.

Setting aside data management, manipulation and analysis are often the focus of a system user's attention. Many users believe, incorrectly, that this module is all that constitutes a GIS. In this portion of the system are the analytic operators that work with the database contents to derive new information. For example, one may need to move data from his/her GIS to an external system where a particular numerical model is available, and then transport the derived results back into the spatial database inside the GIS. This kind of modularity is useful, and at the same time challenging, for the designer of a GIS. It is of particular interest here so far as this book is mainly concerned with analysis.

Product generation is the phase where final outputs from the GIS are created. The output products might include statistical reports, maps, or graphics of various kinds. Some of these products are soft copy images, or transient images on television-like computer displays. Others, which are durable since they are printed on paper and film, are the hard copies. Increasingly, output products include computer compatible material: disks and tapes in standard formats for storage in an archive or for transmission to another system.

Let us emphasize the functionality and configuration of today's GIS (Steiniger and Weibel 2010). According to Steiniger and Weibel, desktop GIS usually serves all GIS tasks which are classified into three categories: GIS Viewer, GIS Editor, and GIS

Analyst, which is another way to reference the five GIS elements discussed above in today's IT environment. Meanwhile, a Spatial Database Management Systems (DBMS) is mainly used to store data, with limited analysis and data manipulation functionality. Increasingly, WebMap Servers are used to distribute maps and data over the Internet. And WebGIS Clients are used for data display and to access analysis and query functionality from Server GIS. Libraries and Extensions provide additional (analysis) capabilities. Examples include network and terrain analysis and to process specific data formats. Finally, Mobile GIS is often used for field data collection.

Several trends of GIS-technology development have been observed:

1. The obvious trend is the continued downsizing of computers. Computers are getting more compact and at the same time more powerful. Despite this trend, the speed of input and output on smaller computers remains a concern for data intensive applications such as digital mapping. The refinement of data-compaction techniques will help to reduce the amount of data to be transmitted and thus increase the throughput.
2. Another obvious trend is that hardware prices will continue to drop, but programming staff salaries will continue to rise. As systems become more complex, more programmers are required to maintain the system. Many of the digital mapping-software products will probably never become consumer goods. The limited market means that the price for these products will continue to remain high.
3. In spite of increasing diversity among computer systems used, there is a pressing need to exchange information among users. This will require standardization of data structure among the different systems—a formidable task until there is a period of stability in GIS developments.
4. Many software functions will be integrated into the computer as firmware. One reason for this is to increase speed of processing. It is likely that workstations dedicated to GIS applications will appear in the future given there is increasing demand for GISs.
5. Over the last couple of years, the number of Web-based GISs have doubled. This trend will likely continue.

When shopping for a software, here are some factors for consideration (Point of Beginning Magazine 2005). They are compiled as a checklist in Table 7.5.

Image Processing Software

Today’s GIS typically processes files in both the vector and raster formats, as surfaced during our discussions above. The latter usually come from earth-monitoring satellites or aerial photography. Such images are processed and integrated with vector files by the GIS for a desired functionality. Explicitly coded spatial information can be extracted from such remotely-sensed data through the use of image processing (IP) software (Vanderzee and Singh 1995). The data can then be combined and compared with other spatially referenced data using a GIS. Since we have already provided a fairly detailed review of GIS software, this section simply contains some supplemental information specific to raster images. It is not meant to be as comprehensive as other surveys reported above. For one, we do not plan to repeat the software selection criteria, as it has been laid out in full under the GIS section.

Table 7.5 FACTORS IN SELECTING A GIS SOFTWARE

- □ Does the software have these operating systems or network support: Network client-server support? Server operating system? Client operating system? Internet server enabled?
- □ Does it offer total solution packages or support these compatible applications: Third party applications designed to run on the software?
- □ Does it have these GIS data administration functions: Multi-user edit locking? Versioning for managing data access by users? Metadata maintenance?
- □ How about these database management supports: Vendor-proprietary DBMS? Relational database management system (RDBMS)? RDBMS spatial data warehouse?
- □ Which of the following native graphic data structure and format does the system employ: Vector-spaghetti? Vector-topologic? Parametric? Three-dimensional? Triangulated irregular network? Grid? Raster image?
- □ Does the software support direct import formats and direct export formats, specifically:
    - What readable import and export formats that do not require translation?
    - Are utility programs bundled with the GIS package for translation of GIS or CAD data to or from another format, including common industry-standard formats like DXF, SIF, DLG or SDTS?
- □ Which of the following utilities does it have for GIS data entry and editing:
    - Board digitizing?
    - Coordinate geometry/precision entry?
    - Electronic survey data import?
    - Heads-up digitizing or on-screen digitizing, where a digitizing station provides a graphical user interface on the screen of a workstation, facilitating the process of tracing outlines from a raster image on-screen?
    - Vectorization (including editing GIS data)?
    - Map rectification (including transformation of coordinate systems and map projections)?
    - Graphic error check/correction?
    - Field data entry?
- □ Does it support these map design and composition functions:
    - Interactive map composition?
    - Modifying map annotation from attributes for custom map displays?
    - Global map symbol change?
    - Automatic creation of thematic maps and legends?
- □ How about these geographic query and analysis functions:
    - Attribute query and selection?
    - Map measurements such as basic distance and area?
    - Address matching?
    - Buffer generation?
    - Point/line-in-polygon analysis?
    - Polygon overlay?

Table 7.5 (CONTINUED)

- Network analysis, such as designing network models based on attributes of network segments?
- Raster document query and access?
- Direct access to other GIS formats?

□ How about these terrain data processing and analysis features:

- Digital-elevation-model generation?
- Contour map generation?
- Three-dimensional display/profile generation?
- Map draping, or to produce an aerial view of a land form, with textured or colored features "draped" over it?
- Slope/aspect analysis? Here "aspect" means the direction in which the land is facing: north, south, east, or west.

□ How about these raster image capabilities: Geometric rectification? Ortho-image generation?

- Image enhancement? Spectral classification?

□ The programming languages that are available for application development:

- Proprietary application development language that is included with the GIS software package?
- Industry standard programming environment (e.g., C++, Visual Basic, Delphi)?

While the Vanderzee and Singh reference is dated 1995, it represents a dearth of scientific information on the subject. Most important, much of what was covered remains valid today. The IP systems covered were designed primarily to manipulate and analyze image data derived from earth-looking satellites or airborne sensors. In this context, IP system capabilities include interactive display, image enhancement, geometric rectification, spatial filtering, image mosaicing, Fourier analysis, radiometric corrections, multivariate analysis, multi-spectral classification, raster-GIS modeling, radar geocoding/analysis, image annotation, and hard-copy output. In the following paragraphs, we will provide more information about the general characteristics, operating environments, supported peripherals, and other capabilities of these software systems.

There has been a dramatic increase in the number of new IP products in the field starting in the late 1980s. The software systems varied widely in their functional capabilities as well as in their price. More than 80 percent of the products included GIS functionality, about a third are considered IP systems, and about 20 percent fell under the categories of automated mapping and facilities management (AM/FM) or computer-aided design.

Technical sophistication of the software products varied widely. About 10 percent of the systems had expert system capabilities; 25 percent had an object-oriented software architecture; 30 percent a spatial index to improve computational efficiencies; 60 percent had an integrated DBMS; and 60 percent also had the capability to link to an external DBMS. In addition to the built-in functions of the systems, 60 percent supported extensions to the system through vendor-supplied macro languages; 40 percent offered linkable libraries for data-structure access; and source code was obtainable for about 20 percent of the systems. Regarding source codes and linkable libraries, C was the language of choice for developing GIS and IP systems. FORTRAN still had a foothold in the 1990s, but C outnumbered it by a wide margin.

About two-thirds of the vendors were able to provide turnkey solutions, including bundled hardware and software, for their clients. Some offered standard packages; others preferred to tailor the configuration to a specific user's needs. More than two-thirds of the software developers offered worldwide software support for their products. Almost all of the remainders offered support for a limited portion of the world. Only a couple indicated that no support was available. About 85 percent of the vendors offered training courses for their clients, about half offered training assistance in the form of tutorials, and about 10 percent offered training videos for their products.

Nearly all the software products offered on-line help. Approximately half of the products had basic on-line help, and about half had a more sophisticated context-sensitive help facility. About a fourth of the products had a full hypertext help facility incorporated into the software system. Some included more than one type of help facility. Documentation was available in electronic form for about half of the software products. Both hard copy and electronic versions were available for many of the products. An English version of the documentation was available for virtually all of the products. A French version was available for less than 10 percent. Even fewer products were offered with documentation in Chinese, Italian, Dutch, Japanese, Spanish, Danish, Greek, and Russian.

The survey showed that the most common operating systems for the various products were UNIX and DOS. Many of the products were offered for more than one operating system. The proportion of systems offered for UNIX and DOS remained pretty much constant. The trend toward windows-based products and graphical user interfaces throughout the software industry was also occurring with GIS and IP products.

More than two-thirds of the products had full graphical user interfaces, about a third had windows-based interfaces; and about a third of the products had simple command-line interfaces. Some products had more than one type of user interface. The most common graphics environments supported by the software products were X-windows for UNIX systems and Microsoft's Windows for PC's. Motif was the most common window manager for UNIX systems. Sun's OpenLook window manager was a distant second, with about half as many products that supported it also supported Motif. A few vendors continued to offer products that supported Sunview and other less common graphics environments, such as Apple Macintosh and Intergraph's Microstation.

The number of installations of GIS, IP, and products increased dramatically over the last couple of decades, mostly in North America and Europe. Developing countries continued to lag behind in the use of these technologies. Many new products were introduced in the last two decades, and the pace of new product introductions appeared to have slowed over the last ten years, reflecting a maturation process. At the same time, the technical sophistication of existing products and new products has increased. The current phase in the GIS and IP software industry is characterized by competition and increased attention to users' requirements.

When shopping for an IP software, here are some factors for consideration in summary:

- □ Function and price: The software systems in the Vanderzee and Singh survey varied widely in their functional capabilities as well as in their price. There was not a direct relationship between functional capability and price.
- □ Technical features
- □ Turnkey systems
- □ Software support
- □ Training assistance
- □ Operating systems
- □ User interfaces
- □ Graphics environments
- □ On-line Help
- □ Documentation

Facility or Site Location Software

In accordance with the facility-location taxonomy identified in this volume, and the review at the beginning of this chapter, it is desirable that a facility-location software be reviewed according to such a taxonomy. A facility-location problem can therefore be identified by a classification scheme consists of five labels, as borrowed from Bender et al. (2002). Accordingly, a software can be located that will satisfy these parameters and render the corresponding solution. Under this scheme, a facility-location problem will carry an identification consisting of five labels:

# facilities/type/assumptions/distance function/objective function

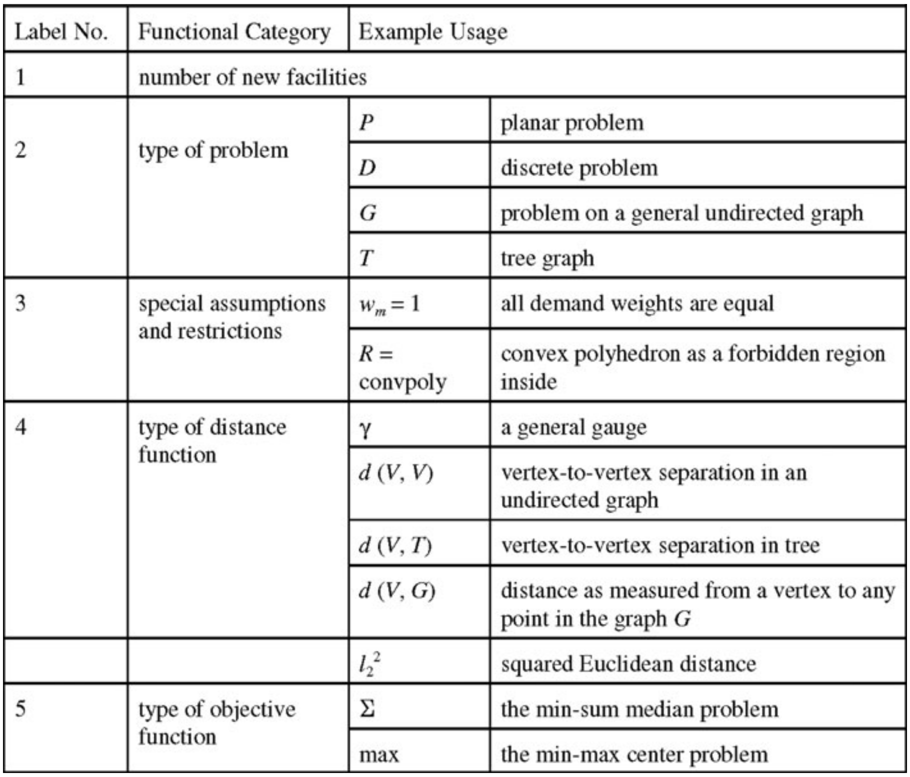
Each of these labels is defined in detail in Table 7.6. As an example, an identification of $3/G/w_m = 1/d(V, V)/\Sigma$ would suggest that the software will solve a three- facility (3) median-location (Σ) problem on an undirected network graph (G) with inter-nodal distances ($d(V, V)$), where each demand node carry the same weight ($w_m = 1$). For each of the five labels, the label is indicated by a * if no special specification is given.

Some of the desirable features of a facility-location software are listed below:

- □ graphical user interface
- □ linking with professional graph-editing and graph-drawing programs
- □ linking with professional data-management programs
- □ A suite of callable libraries
- □ User manual
- □ Interface with GIS
- □ Interface with mathematical-programming software (such as CPLEX)
- □ Interface with supply-chain management software (such as the SAP Enterprise Resource Planning software)

For data management, C++ Standard Template Library (STL) is a powerful library containing basic data structures and algorithms. With respect to GIS integration, it is desirable to have a completely memory-resident data exchange in order to increase execution speed.

Table 7.6 CLASSIFICATION OF LOCATION SOFTWARE



Label No.	Functional Category	Example Usage	
1	number of new facilities		
2	type of problem	*P*	planar problem
		D	discrete problem
		G	problem on a general undirected graph
		T	tree graph
3	special assumptions and restrictions	$w_m = 1$	all demand weights are equal
		R = convpoly	convex polyhedron as a forbidden region inside
4	type of distance function	γ	a general gauge
		$d\ (V, V)$	vertex-to-vertex separation in an undirected graph
		$d\ (V, T)$	vertex-to-vertex separation in tree
		$d\ (V, G)$	distance as measured from a vertex to any point in the graph *G*
		l_2^2	squared Euclidean distance
5	type of objective function	Σ	the min-sum median problem
		max	the min-max center problem

SOURCE: Adapted from Bender et al. 2002.

To date, facility location is a relatively narrow field that has yet to attract commercial software vendors to enter the market. For this reason, most available software is developed by universities and research organizations. We will survey some of these software packages later in this chapter when we discuss public-domain software (or "freeware"). For the time being, the above discussion can be thought of as a prescription for future software vendors when they decide to enter the market. Meanwhile, the related application in facility layout and supply-chain management commands a much larger market than simply facility location. Google offers the SketchUp software for facility layout, including three dimensional displays. The SAP Advanced Planner and Optimizer (SAP APO) is an integrated software application for supply-chain planning that "backs" into facility location. Obviously, the SAP APO is not designed to solve facility-location per se.

Routing Software

Another spatial analytic tool is vehicle routing, which is a component of supply chain and location-based services. Here, we are more concerned with supply-chain applications than providing an individual commuter with mobile navigation. With supply-chain application in mind, routing software companies are offering creative ways to integrate computer, communication and location technologies with algorithms and software (Partyka and Hall 2010). New data sources have recently become available, including a more complete commercial-road database, and true historical traffic data based on real travel times. There is an explosion of map-data attributes and capabilities. In the next year or two, we will have predictive travel speeds for road segments down to 15-minute intervals. We are now seeing stronger connectivity between routing software's traditional functions—that of assigning stops to drivers and placing them in an optimal sequence—with on-the-road navigation. A printout listing turn-by-turn directions is being replaced by voice commands and dynamic map displays from a driver's phone or navigation device. Mobile phones are changing the industry in a big way because they allow real-time data capture, which in turn enables real-time re-optimization of the operation. This is of particular interest because dynamic scheduling algorithms can make use of these data to increase operational efficiency.

Desktop-based routing is going away. Instead, people want Web-based solutions, so all parts of the organization can have visibility. This is now often accomplished through the Software as a Service (SaaS) model, whereby the software vendor generates solutions and manages data from their own servers. The routing software surveyed by Partyka and Hall in 2010 provide a common set of basic capabilities:

1. geocoding addresses, i.e., locating the latitude and longitude by matching the address against data contained in a digital map database;
2. determining the best paths through street networks between pairs of geocoded points;
3. solving vehicle routing problems, entailing an assignment of stops to routes and terminals, sequencing stops and routing vehicles between pairs of stops; and
4. displaying the results in both graphical and tabular forms in such a way that dispatchers can guide the solution process and communicate results to drivers, loaders and other personnel.

More than half of the products offer some capability for real-time routing, which could come in the form of real-time vehicle re-routing or real-time stop scheduling. Six vendors—Appian, Descartes, FreshStart Logistics, MJC2, SAITECH and UPS Logistics—have the ability to incorporate real-time traffic, which is now more widely available in major cities. The collective capabilities enable a fleet to reschedule in response to customer requirements, vehicle delays or traffic conditions.

Whereas vendors generally claim that their products are designed to serve a broad range of applications, most specialize in an industry sector. Specialization is largely driven by interface requirements—both in terms of presenting information in a manner that is useful to the target user and in terms of interfacing with business software systems and hardware devices. Police, taxi and emergency vehicle dispatch, for instance, each demand special requirements that differ from those of private fleets. They fall in the realm of niche markets, even though in theory they are just variations of vehicle routing. Nevertheless, here are the other factors that should be considered in choosing a software package. They are compiled as a checklist in Table 7.7.

B. Developmental Geospatial Software in the Public Domain

Supplementing commercial software are those that are available in the public domain, free for users to access. However, one should distinguish between

Table 7.7 FACTORS IN SELECTING A VEHICLE ROUTING SOFTWARE

□ Platforms supported: Windows, Linux, Unix, Mac OS, Application as Service (system utility to run applications as Windows services);
□ Maximum size of problem solvable by the system: Number of stops, number of vehicles, number of terminals;
□ Recommended hardware, processor speed, memory, hard disk space;
□ Performance: Computation time? What types of algorithms are employed (open ended)? Are approximations used to reduce computation time?
□ Routing functions: Node Routing, arc routing, real-time re-routing, real-time stop scheduling, daily routing, route planning and analysis? Incorporate real-time traffic information?
□ Price information: Quotation for, say, a single site license for 50 routes? Does license fee include map for one region? What brand of map is provided? Installation support cost, say, in $/hour? Typical support hours needed for installation, assuming a 50-routes system?
□ GIS capabilities: Displays routes and stops on maps? Can edit routes with drag and drop? Geocodes stops from addresses?
□ Solution algorithm: Does system accept soft time windows? If so, how are soft time windows specified?
□ Product is available as part of a suite that provides these services: On-board electronic display? Wireless messaging to driver? Real-time vehicle tracking? Barcode scanner? Supply-chain-management software (e.g., inventory management)? Custom order process?
□ Features: Individual driver assignment? Turn-by-turn route instructions? Automatic forecasts of delivery? Load manifests? Loading plan for truckload? Weather forecast information display?
□ Types of fleets that currently use this product: Local pick-up and delivery? Longhaul less than truckload? Long-haul truckload? Courier? Buses? Taxis? Service fleets? Emergency service (police, fire, etc.)?
□ Other special features

- Recent innovations in system?
- Has your routing software been integrated with either cell phone or PDA technology?
- Have you developed other software innovations, such as use of social networking for information sharing?

□ New features that address sustainability/green requirements?
□ Number of Companies Using Software?
□ Most Significant Installations?

Open-Source Software vs. "No Cost" Software. Open-Source Software allows the user to make functional changes. It allows the user community the freedom to maintain and enhance the software when the original developers are no longer available. Other "no cost" software often does not distribute source codes and thus has "locked functionality." These "no cost" software can be withdrawn by the developer at any time, leaving the software in a "frozen state." Where information is available, we note this distinction in the detailed software survey in Chapter 8.

1. Open Source Software. Here are the stipulations under an Open Source (OS) software. Users can freely use the software at home or at the office and use it on whatever computers they want to. Users can give the software to whoever they want to. Users can make programming changes to the software, adding features that may be missing or even change the way some features work. Usually the licenses for this software prevent the first party user from restricting the freedom of the second party user who may receive the software from the first. There are a large number of different open-source licenses. The most popular is the GNU Public License (GPL), where the operating system GNU stands for "GNU's Not Unix!". The Lesser GNU Public License (LGPL) allows mixing of proprietary and open-source components without having to release the source of the proprietary components. The pertinent components under the LGPL must still be "source available," including changes or improvements. There are organizations promoting OS software. The Open Source Geospatial Foundation (OSGeo) began in 2006 with nine projects forming the foundation for promoting OS geospatial software. Their project sponsors must be granted an Open-Source-Initiative-compliant license in order to be an OSGeo member. They are encouraged to use an LGPL or similar license so that libraries can be reused by non-GPL projects. OSGeo provides resources (such as funding and infrastructure) to member projects. It provides support for the use of OSGeo software in education. It operates the annual OSGeo conference, and it promotes the use of all OS software in the geospatial industry. In parallel, there is a recent movement toward open-source optimization software (among others). A prominent example is the Computational INfrastructure for Operations Research, or COIN-OR for short. The project is also managed by a non-profit foundation.

2. Freeware. FreeGIS.org is a comprehensive source of information about free geospatial software, geo-data and documents. This is an excellent location to consult when searching for free geospatial software. However, please be aware that there is a wide range of software maturity. There are two primary sources of such "freeware" packages. The first is the U.S. government (and other international governments that have the same policy). The second are the research and educational institutions, developed mainly for pedagogy. While the commercial market tends to be driven by demand, the government and research organizations are generally at the periphery of the marketplace, rendering them to be more experimental in software development.

In recognition of its importance, an increasing pool of public-domain, state-of-the-art, spatial-information software has become available from research and educational institutions. Obviously, there are commercial vendors who venture into the development of state-of-the-art software. Likewise, the government may develop software that has been available in the commercial sector. For example, a better-known government-sponsored software is GRASS, a GIS that has been available for quite some time. Standing for Geographical Resources Analysis Support System, GRASS is a public-domain raster GIS, a vector GIS, an image-processing system, and a graphics-production system. It is extensively used at government offices, universities, and commercial organizations. It is written mostly in C for UNIX.

There is a rich repository of developmental freeware from university campuses. These range from comprehensive packages to more specialized tools. The University of Tennessee at Knoxville, for example, has assembled tools from environmental assessment fields into an effective problem-solving environment. These tools include integrated modules for visualization, geospatial analysis, statistical analysis, human-health risk assessment, ecological risk assessment, cost/benefit analysis, sampling design, and decision analysis. It is clear that there are similar packages available elsewhere.

Spatial Statistics

Rey and Anselin (2006) reported some software development efforts on university campuses and research institutions. Many of them provide a tight coupling of spatial and nonspatial data representation and queries. This coupling relies on quadtree-style indexing strategies and originally focused on two-dimensional spatial objects, such as county boundaries and river systems. A third dimension can be incorporated through a new object type, the *isosurface*, a three-dimensional surface that represents points of a constant value. This results in a powerful technique for performing clustering and windowing. Ideally, the software will allow dynamic exploration of areal data measured over multiple time periods, such as analyzing data on sudden infant death syndrome (SIDS) in North Carolina [as cited in Rey and Anselin (2006)]. Systems can include mapping and geo-visualization to spatial autocorrelation analysis, multivariate exploratory data analysis, and finally confirmatory spatial regression analysis. Here, the word *confirmatory* is used to distinguish it from *exploratory* analysis.

Some systems include two classes of transportation network data, those that occur on a network (e.g., traffic accidents) and those that occur in proximity to, but not on, a network (e.g., restaurants in an urban area which may have entrances on multiple streets). Analyses include hot-spot detection, spatial interpolation, and journey-to-crime computation. The systems allow the creation of a wide range of spatial weights from various input formats, the computation of higher order weights and the construction of spatially lagged variables. It also implements the numerical procedures needed for spatial data analysis. These include exploration (such as rate smoothing and outlier detection), description of global spatial autocorrelation (such as *Moran* and *Geary statistics*), and spatial regression (using maximum likelihood and method of moments estimation).

Location Theory

As far as facility location, Bender et al. (2002) developed LoLA - Library of Location Algorithms, following the taxonomy as outlined under the "Facility or Site Location Software" subsection. They tied LoLA to the ArcView GIS, the SAP software as well as CPLEX. Ottensmann (2000) used spreadsheet optimization to teach facility location and spatial interaction, with accompanying Website showing the Excel spreadsheets implementing these models. In a more dated, yet more comprehensive fashion, Ottensmann (1985) presented educational BASIC programs to perform many of the functions documented in this book:

1. Trend projection models
2. Population cohort-survival model
3. Economic base model
4. Shift-and-share model
5. Input-output model
6. Single-constrained gravity model
7. Double-constrained gravity model
8. Facility-location-on-a-plane model
9. Facility-location-on-a-network model

Meanwhile, Daskin disseminates the following software to accompany his book (Daskin 1995).

1. SITATION: Facility Location Software
2. Spreadsheet for the Traveling Salesman Problem
3. Spreadsheet for Facility Location Problems
4. Time Dependent Queueing Analyzer

While the SITATION software is amply documented in Daskin's book, the following enhancements have been made to the original version of the program. The programs are entirely menu-driven and run under Windows 95 and later versions of Windows. SITATION now solves five classes of location problems, including *p*-median, *p*-center, set covering, maximal covering, and un-capacitated fixed-charge. SITATION includes branch-and-bound capabilities to allow the user to obtain very tight solutions. Additional mapping capabilities have been added. SITATION allows the user to zoom in on portions of the tradeoff curves and maps. The software now allows the user to specify alphanumeric (text-based) node names. SITATION solves the covering-median tradeoff problem using the weighting method. As posted on the Daskin webpage, the newest version of the software will solve problems with up to 300 nodes.

3. Software Accompanying This Book. Some developmental software is included in the CD/DVD that accompanies this book. This set of software was developed specifically in support of this volume and the companion Chan (2005) volume. The spatial analytic applications range from location theory to image processing. These philosophies are followed in the preparation of software on this disk:

1. In order to provide the widest dissemination possible over time, all files are ASCII-text files, PDF files, input file for a generic mixed-integer program (MIP) or MATLAB code--representing some commonly available media in the community. Other than the generic MIP or MATLAB code, all software executes under the DOS operating system. For most of the codes, both source codes and executable codes are given—mainly for the ease of execution and modification by the users.
2. We strive to provide standalone programs that do not require supporting software, including language compilers. All programs are self-contained and they have been developed or refined by the author and his associates. For extended use of some of the programs, references are made to optional supporting software, such as using the freeware OCTAVE as a replacement for MATLAB.
3. Sample datasets are provided to allow demonstration of the software. While "toy" problems are often used for introduction, most of these data are drawn from real-world case studies which are discussed in the main body of this book and Chan (2005).

Our spatial-analytics software is organized into seven folders:

- □ STATEPRK—A folder that contains a location model based on activity derivation-allocation.
- □ SPANFRST—A folder that houses a heuristic location-routing program for small-package deliveries.
- □ RISE—A folder that includes a heuristic location-routing model for scheduled passenger-transportation service.

- □ SPACEFIL—A folder that presents a heuristic multiple traveling-salesmen program.
- □ LOWRY—A folder that has the traditional Lowry land-use model, based on economic-base activity derivation and gravitational allocation.
- □ YICHAN—Bearing the names of the developers, this is a folder that debuts a disaggregate/bifurcation implementation of the Garin-Lowry model, a successor to the Lowry model.
- □ PATTERN—This folder contains the K-MEDOID algorithm, which is a classification software for a grid of gray values (such an image). It also contains a input text file for an MIP code in long-hand equation format.
- □ SPACE—This folder contains the TS-IP program, which is an image-processing program that loads and manipulates digitized satellite images.

A complete User's Manual accompanies these software files on the CD/DVD. While datasets are provided for each model, a number of satellite images of the U.S. are also included in the IMAGEFILES folder for experimentation through the image processing programs K-MEDOID and TS-IP.

C. Selecting a Software: The Case of GIS

Having reported available software on the market, how does one select the software that is most appropriate for his or her applications. Let us run through this exercise using GIS as an example. Many organizations are faced with the decision to acquire or to upgrade a GIS (McCrary et al. 1996). This is to be performed in an environment where the technology is rapidly changing. Following our discussion of GIS software earlier in this chapter, a comprehensive set of criteria was listed. From the list, let us say the potential user decided that selecting a GIS in his/her organization typically involves answering these questions:

- □ Do you want the ability to access database and graphics from the same package?
- □ Do you want the ability to integrate between software packages, including between GIS and packages which perform analysis?
- □ Are you willing to pay the price for integration?
- □ Do you have the expertise to use GIS software?
- □ Do you want the ability to conduct spatial analysis such as facility location and land use?
- □ Is the acquisition or upgrade an efficient use of GIS in your organization?
- □ Will your system be networked in the future?
- □ Is a topological database needed?

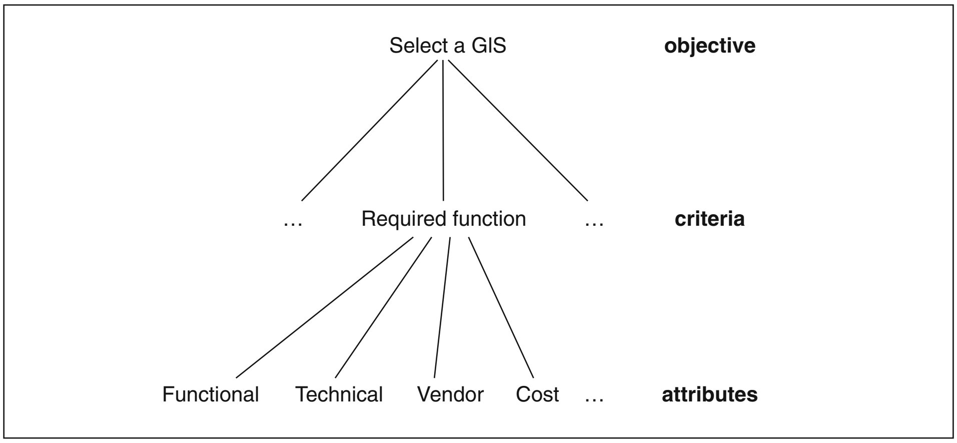
The potential user agreed that the ultimate driving force behind a GIS selection has to be the problems to which the software is applied. Some of the basic applications in his/her organization can be enumerated below:

1. **Geographic data collection and production:** This refers to the fundamental GIS function of collecting geographic data for the purpose of building both spatial and non-spatial databases, as described in Chapter 6 and in the current chapter.

2. **Facility and asset management:** This means locating, counting, analyzing, and/or reporting on the distribution of facilities and assets that are on, below, or above the earth, for the purpose of inventorying them for usage.
3. **Map and chart publishing:** This includes producing and publishing maps and charts for the purpose of direct distribution or documentation.
4. **Resource allocation:** This means the fundamental task of analyzing, allocating, and reporting on a resource's location, quantity, quality, and/or movement, for meeting certain economic, financial, political, and social criteria of specific interest to her organization.
5. **Network analysis:** This refers to analyzing, scheduling, routing, and/or reporting the flow of people, goods, or services through the organization's network for best usage.
6. **Site selection:** This is the core function of selecting and reporting on the desirable site based on a set of imposed criteria for optimizing location.
7. **Surface and sub-surface assessment:** This is concerned with modeling, analyzing, and reporting on the natural geophysical phenomena occurring on or below the surface, for understanding, preserving, or exploiting such phenomena.
8. **Tracking and monitoring:** As a bottom line, the potential user worries about recording, analyzing, and reporting activities over time for understanding the occurrences, and/or for developing complementary or corrective responses.

It is obvious that depending on the function the GIS is required to perform, a very different package(s) may be selected. Among other techniques, multi-attribute utility-theory (MAUT) can be the scientific method for evaluating and selecting GIS software. In MAUT, the elements of a decision-making problem are broken down into a hierarchy of objectives, criteria, and attributes. An example of such a hierarchy is shown in Figure 7.1. For a particular application and a GIS under consideration, the following multi-attribute

Figure 7.1 HIERARCHY OF OBJECTIVES IN GIS SELECTION



SOURCE: McCrary, Benjamin, and Ambavanekar (1996). Reprinted with permission.

utility-function may yield the necessary metric v for evaluation: $v = v\ (f', t'', s', c)$ where f' is the functional-attribute score, t'' is the technical-attributes score, s' is the vendor score, and c is the price—all normalized between a scale from 0 to 1, where a larger value is more desirable. Examples of technical attributes may be user-friendliness, performance and expandability; while examples of vendor attributes may include experience, reputation, quality of documentation, quality of support etc. Having defined these attributes, a simple addtive value-function may be justified, consisting of $v = w_f f' + w_t\ t'' + w_s\ s'$ where w_f, w_t and w_s are weights assigned to the respective attributes by the stakeholders. Obviously, there are other forms of value functions, depending on whether one wants an ordinal ranking or a cardinal ranking of GIS packages. The key point is that the calibration and implementation of such value functions for GIS selection are situation-specific. Another important point is the MAUT allows for explicit tradeoffs between various criteria and attributes. This is often more important than the single metric v that may fall out of such a procedure.

In Chapter 8, entitled "A Software Survey of Analytics and Spatial Information Technology," we have provided a screened list of commercial and public-domain software. As documented in Chapter 8, the list represents some "popular" software based on the criteria that have been set up for each application, whether it be simulation software, statistical software and so on. Notice the screened list does not represent endorsement on the part of the author. Where necessary, the reader is encouraged to go beyond the list depending on his/her particular needs, which is the reason for the discussions in this section to begin with.

IV. SPATIAL INFORMATION TECHNOLOGY: LOOKING AHEAD

Recent events have deepened our conviction that many human endeavors are best described in a geospatial context. This is evidenced in the prevalence of location-based services, as afforded by the ubiquitous cell-phone usage. It is also manifested by the popularity of such Internet geospatial IT tools such as Google Earth and GPS vehicle navigation. As we commute to work, travel on business or pleasure, we make decisions based on the geospatial information provided by such location-based services. When corporations devise their business plans, they also rely heavily on such geospatial data. By definition, local, state and federal governments provide services according to their respective geographic boundaries. With geospatial information at one's fingertips, governing bodies wish to see how one can use it to ensure public safety and security, which is a most stringent requirement, in as much as the relevant information has to be available to make split second decisions. One estimate suggests that 85 percent of data contain spatial attributes. This is not even counting the interpretation that Internet domain names are the "real estate addresses" of the 21st century. Nor does it include the virtual reality world, in which one immerses herself in a "second world" without really being there.

A. Spatial Information Technology

Scientifically speaking, decision-making is based on information and intelligence extracted from data. Manipulating spatial database systems has been perceived as a distinct and specialized field of information technology (Yeung and Hall 2007). There are simply quite a few unique characteristics of spatial data that are distinct from conventional databases. These include the distinct nature of spatial data representation, the use of map projection and coordinate systems, the nature of spatial statistics, and the pervasive need to respect cartographic presentation. Acquisition of a working knowledge and mastery of these topics requires years of education and practical training, as witnessed by the extensive discussions in this text. There are always professional spatial-data users in government, business, and academic research who call for spatial analytical functions that conventional commercial database systems cannot fulfill.

Recently, the inclusion of spatial-data handling in mainstream-database software has grown consistently and the emergence of a healthy open-source geospatial software community has meant that the mainstream and spatial database worlds have been converging. The integration was driven by several subtle but interrelated factors:

1. Advances in computer hardware, software and standards, have helped to overcome the longstanding incompatibility between spatial and non-spatial data representation and processing.
2. The advent of the Internet and networked computing has stressed standardization, interoperability and usability, which have effectively removed the boundaries and barriers between spatial and other branches of IT.
3. There is a growing demand for novel and sophisticated spatial applications, as witnessed by location-based services. This has forced spatial-database software vendors to look for methods and tools outside the traditional realms of GIS technology.
4. The growing recognition of the importance of spatial information as a commodity with value for modern society and the resulting business opportunities have motivated mainstream IT companies to enter the spatial database marketplace.

We mentioned that there are three types of software: commercial, open source, and general public-domain codes. A common belief is that brains, ideas and algorithms originating in academic spatial analysis and migrate to the private sector when the market develops. This is an overly simplified view. The two communities sometimes come together when this type of migration is mirrored in the infusion of support to academic research projects from the private sector. In fact, there are numerous examples of companies engaging with open source projects to their benefit.

Let us now turn to the open source software. Considering that the open source movement is only a decade old, its footprint on the world of spatial IT is impressive (Rey 2009). At the same time, a closer examination suggests that the contributions have been most heavily concentrated on spatial data and traditional GIS functionality, while open source projects in the areas of advanced spatial analysis, statistics, spatial econometrics and spatial modeling tend to be much less prevalent. These areas sit at the top of the spatial analysis research pyramid and reside mostly in the academic and research community. We have seen examples of this in our discussion of "freeware" in the Software section in the current chapter.

While the potential for cross-fertilization between the open source movement and academic research on spatial analysis is promising, it is by no means inevitable. There are a number of factors that prevent such cross fertilization. In the academic and research community, the value of a software manual is preempted by that of a refereed journal article. This often results in codes that are poorly documented, preventing them from broader dissemination. Also, there is a disconnect between an open source academician's contribution and the attribution s/he is entitled to. The researcher risks a loss of attribution if the original source code were shared openly with the broader scientific community.

Open source can play a vital role in today's new research era. Relying on open standards and programing frameworks facilitates the integration of specialized application programs into scientific middleware. Open source code as a way to implement integrated models provides a transparency that can facilitate communication between scholars from different domains. This new research era is also characterized by the growing complexity of the research questions being posed. Increasingly researchers are relying on numerical simulation for results, as closed-form solutions are not available for emerging research questions. This, in turn, is blurring the roles of software developer and scientist, as success of the latter will increasingly require competence in programming.

A GeoPortal is a type of web portal used to find and access geospatial information and associated services (such as display, editing, analysis) via the Internet. Following this philosophy, Ferreira et al. (2010) outlined the framework and implementation of a flexible, loosely coupled information infrastructure to facilitate collaborative research on spatial analytics. The framework combines off-the-shelf open source applications such as Apache, PostgreSQL, Mapserver, OpenSSL, and MediaWiki, with proprietary tools such as ArcGIS Server and Flex, and uses minimal custom code to provide web services for distributed modeling and realistic evolution of data sharing. Here is the hierarchy of such an architecture, going from the basics at the top of the list to the more ambitious at the bottom of the list:

- □ File transfer
- □ File transfer with search
- □ Dynamic geospatial infrastructure
- □ Distributed dynamic infrastructure
- □ Geospatial compute cloud

Four layers are envisaged in this proposed architecture: analytic layer, presentation layer, middleware layer, and data layer. It is clear that such an architecture represents merely a first step of such an effort. These critical issues need further exploration (Ferreira et al. 2010):

1. decomposing work flows into modules that are meaningful and "thin" enough for one group's focus while having stable and well defined inputs and outputs,
2. balancing the benefits of automation and process management tools with the labor and maintenance cost of customized code,
3. training researchers and workgroups to utilize the information infrastructure effectively, and
4. addressing "version skew" issues as components of the GeoPortal mature.

According to Rey (2009), the most successful open-source projects are those not only with excellent code bases but thriving communities of users and developers. Cross-fertilization will come only when the number of producers of open source code grows along side the consumers of such projects. However, the magic of the open source movement lays not in its fascinating social dynamics, but in the promise of new ways to organize science and heighten the pace of knowledge discovery.

B. Going Beyond

Recent advances in wireless communication technologies are now adding a new dimension to technology integration that plays a pivotal role in spatial and mainstream IT integration. Wireless communication devices such as mobile phones, pagers and computers have decreased in size, weight and cost, and have increased in functionality, portability, security and reliability. The advent of Tablet PCs is particularly noteworthy. Tablet PCs differ from the other mobile computers in these important ways:

1. Full-feature operating system such as Windows XPTablet PC Edition allows Tablet PCs to run any existing Windows applications.
2. Because of their compact size, light weight, high capacity memory, pen-based input and most importantly, long battery life, Tablet PCs are more mobile than other types of mobile computers.
3. Tablet PCs use a pen device that replaces the traditional mouse. A user would tap and press on the screen with this device to interact with applications. The interface allows a user to write on the screen using *digital ink*. Each ink stroke and its color, width and attributes can be edited and stored just like traditional graphics and text.

Tablet PCs have overcome most of the limitations of display screen size, storage capacity and processing power imposed by laptops, PDAs and hand-held computers. Digital ink allows Tablet PCs to be used not only for the retrieval, processing and display of spatial information, but also for real-time capture and editing of file data through the use of GPS devices, electronic field survey equipment and handwriting. The processing power and data storage capacity of a Tablet PC are comparable to a desktop computer, hence it can be used as a "thick" client in a typical client/server computing environment. As such, it is able to minimize the amount of data traffic that would otherwise be required in a thin client architecture.

The Global Positioning System (GPS) has made available accurate locational information, where the two second-generation global satellite-navigation systems in operation today are GPS II and GLONASS. The abilities of a mobile computer to determine its own position and make use of this locational information in spatial data processing has allowed a large number of spatially-enabled applications to be developed. Looking toward the future, GPS III is probably best characterized as a third-generation system where the focus is on improvement and modernization. These will result in real-time supply of driving directions, emergency response locations, traveler information, advertising and marketing and real-time environmental data collection. The advent of the devices that streamline such location-based services represents one other major innovation of spatial database development in the last decade.

Google, Microsoft, and Yahoo are racing to transform online maps into full-blown browsers, organizing a diversity of information (Chan 2009). Google Earth combines satellite imagery, maps and the power of Google Search to put the world's geographic information at one's fingertips. Since its debut in summer 2005, Google Earth has received attention of an unexpected sort. Officials of several nations have expressed alarm over its detailed display of government buildings, military installations and other sensitive sites within their borders. Beyond Google, Globalsecurity.org has images of nuclear test sites and military bases in much sharper focus than can be found on Google Earth. The company was asked by the National Geospatial-Intelligence Agency, an arm of the U.S. Defense Department, to remove from their site some of the maps of cities in Iraq, which was at war during that time. Without implications of endorsement or dis-endorsement, however, the incident—among others—was a classic example of the futility of trying to control information.

Very briefly, let us continue to provide some food for thought in the exciting subject of mining geospatial data. The potential application in safety and security is endless (Rouch 2007). A systems-biology graduate student Andrew Hill and colleagues at the University of Colorado published a KML file in April 2007, with a grim animated time line showing how the most virulent strains of avian flu jumped from species to species and country to country between 1996 and 2006. What if you could model a Europe where the sea level is 10 feet higher than it is today, or walk around the Alaskan north and see the glaciers and the Bering Strait the way they were 10 years ago or in the prehistoric past when the earth went through dramatic climate changes? Then perceptions around global warming might change one way or another. While we are laying out a future agenda in this chapter, much of the technology is here already. Digital globes are gaining in fidelity, as cities are filled out with three-dimensional models and old satellite imagery is gradually replaced by newer high resolution shots. Moreover, today's island virtual worlds will only get better, with more-realistic avatars and settings and stronger connections to outside reality. Map algebra for cartographic modeling was introduced by Tomlin (1990). In parallel, Ritter et al. (1990) introduced image algebra for image processing. Perry, Sheth, Arpinar, and Hakimpour (2009) proposed geospatial and temporal semantic analytics. In this context, semantic refers to the meaning of data rather than its syntax or structure. If one can understand and process data on using map algebra, image algebra, or semantic analytics, s/he can achieve a higher level of automation, integration, and inter-operability. It is a fascinating world for experimentation, with potential applications to data integration and information-quality assurance. While data integration remains a main focus, the latter is becoming a topic for increasing attention, inasmuch as pertinent decisions can only be made on quality data.

V. EXERCISES

Self-Instructional Module: LINEAR PROGRAMMING PART 2 - SOLUTION ALGORITHM (to be found on the attached CD/DVD)[7]

In the current chapter, entitled "Analytics and Spatial Information Technology," some solution software packages are presented. An example is optimization software, in which linear programming plays a central role. In the solution of many optimization problems, such as mixed integer programs, the algorithm is often based on solving repetitive linear programs (LPs). It should be obvious to the reader by now that the graphical method for solving linear programs—such as shown in Figure 4.7—is for illustration only, and is limited to models of two variables such as shown in Illustration (1). The models such as Illustration (3), with three or more variables, are not easily sketched on two-dimensional graph paper. Thus, an algebraic technique is needed to solve LPs with numerous variables and equations. Also, such an algebraic technique is conducive to computer solution. One algebraic technique for solving linear programs is called the simplex algorithm. It was proposed by George Dantzig in 1947, and its theoretical foundation was established in 1948 by Gale, Kuhn and Tucker in the working paper "Extremum Problems with Inequalities as Subsidiary Conditions." For more information, the interested reader is referred to the classic volume, *Linear Programming and Extensions*, by George Dantzig, Princeton University Press, 1963.

The simplex algorithm illustrated in this module is limited to LP models formulated in a particular format. In real life situations, LP models usually assume many different forms, and they have hundreds and thousands of variables and constraint equations. Thus, a more advanced understanding of LP and computer programs are essential in solving these LP models. Chapter 4 (entitled "Prescriptive Tools") and Appendix 4 of this text (entitled "Optimization Schemes") will provide more depth in linear programming. Many available software packages are quite efficient in handling LP models of large size. For convenience, the author has included a software survey in both the current chapter and Chapter 8.

ENDNOTES

[1] The "Moving Picture Experts Group" (MPEG) is a working group of experts that was formed by International Organization for Standardization and International Electrotechnical Commission to set standards for audio and video compression and transmission.

[2] An NLP is a separable program if its objective function and all constraints consist of separable functions, i.e., $f(x) = \Sigma_j^n f_j(x_j)$ and $\Sigma_j^n g_{ij}(x_j) = b_i$ for $i = 1, \ldots, m;$ and all x_j are non-negative variables bounded above, i.e., $0 \le x_j \le m_j$ for some m_j, $j = 1, \ldots, n$.

[3] More precisely, let **C** be a subset in a real (or complex) vector space. If $\boldsymbol{\lambda} \mathbf{C} \subset \mathbf{C}$ for any real $\boldsymbol{\lambda} > \mathbf{0}$, then **C** is called a cone.

[4] For example, the matrix $\mathbf{M} = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$ is positive definite. For a vector $z = (z_1, z_2)^T$, the quadratic form is $z^T M z = z_1^2 + z_2^2$. When the entries z_1, z_2 are real and at least one of them nonzero, the quadratic form is positive.

[5] Constraint programming (CP) is an emergent software technology for effective solution of large, particularly combinatorial, problems (Barták 1999). Not only it is based on a strong theoretical foundation but it is attracting widespread commercial interest as well. CP is a programming paradigm where relations between variables are stated in the form of constraints (Wikipedia 2011). Constraints differ from the common primitives of imperative programming languages in that they do not specify a sequence of steps to execute. Instead, they specify the properties of a solution to be found. This makes CP a form of declarative programming. The constraints used in CP are of various kinds: those used in constraint satisfaction problems (e.g., "A or B is true"), those solved by mathematical programming solvers (e.g., "$x \leq 5$"), and others. The important feature of constraints is their declarative manner, i.e., they specify what relationship must hold without specifying a computational procedure to enforce that relationship. The idea of CP is to solve problems by stating constraints (requirements) about the problem and, consequently, finding solutions satisfying all the constraints. Constraints are usually embedded within a programming language or provided via separate software libraries. As a computer language, the magic of CP is that the user states the problem, the computer solves it.

[6] Global optimization is a branch of applied mathematics that deals with the optimization of a function or a set of functions to some criteria (Wikipedia 2010). In real-life problems, functions of many variables have a large number of local minima and maxima. The objective of global optimization is to find the best solution in the presence of multiple local optima (Pintér 2011). Formally, global optimization seeks global solution(s) of a constrained optimization model. The most common form is the minimization or maximization of one real-valued function in the parameter-space. There may be several constraints on the solution vectors. To formulate the problem of global optimization, assume that the objective function and the constraints are continuous functions, the component-wise bounds related to the decision variable vector are finite, and the feasible set is nonempty. These assumptions guarantee that the global optimization model is well-posed since the solution set of the global optimization model is nonempty. Finding an arbitrary local optimum is relatively straightforward by using local optimization methods. Finding the global maximum or minimum of a function is much more challenging and has been practically impossible for many problems to date.

[7] The answer to this Module is attached at the end of this textbook.

REFERENCES

Ahituv, N.; Berman, O. (1988). *Operations management of distributed service networks: A practical quantitative approach.* New York: Plenum Press.

Barták, R. (1999). "Constraint programming: in pursuit of the holy grail." *Proceedings of the Week of Doctoral Students* (WDS99), Part IV, June: 555–564.

Benabdallah, S.; Wright, J. R. (1992). "Multiple subregion allocation models." *Journal of Urban Planning and Development* 118, No. 1:24–40.

Bell, P. C. (2008). "Riding the analytics wave." *OR/MS Today*, 35, No. 4.

Bender, T.; Hennes, H.; Kalcsics, J.; Melo, M. T.; Nickel, S. (2002). "Location software and interface with GIS and supply chain management." In *Facility location: Applications and theory*, edited by Z. Drezner and H. W. Hamacher. Berlin and New York: Springer-Verlag.

Bennion, M. W.; O'Neill, W. (1994). "Building transportation analysis zones using GIS." *Transportation Research Record* No. 1429: 49–56, Transportation Research Board, Washington, D.C.

Bhadury, J.; Tovey, C. (2010). "An improved implementation and analysis of the Diaz and O'Rourke algorithm for finding the Simpson point of a convex polygon." *International Journal of Computer Mathematics* 87:244–259.

Buckshaw, D. (2010). "Decision analysis software survey." *OR/MS Today* 37, No. 5, October.

Buede, D. M. (undated). Decision/risk analysis software: Survey for trade studies. Working Paper. Department of Systems Engineering, George Mason University, Fairfax, Virginia.

Chan, Y. (2005). *Location, transport and land-use: Modelling spatial-temporal information*. Berlin and New York: Springer.

Chan, Y. (2009). "Visualization and ontology of geospatial intelligence." *In Data Engineering - Mining, Information and Intelligence*, edited by Y. Chan; J. Talburt; and T. Talley. New York and Berlin: Springer.

Daskin, M. (1995). *Network and discrete location: Models, algorithms, and applications.* New York: Wiley.

Davenport, T. H.; Harris, J. G. (2007). *Competing on analytics: The new science of winning.* Cambridge, Massachusetts: Harvard University Press.

Ferreira, Jr., J.; Diao, M.; Zhu, Y.; Li, W.; Jiang, S. (2010). "Information infrastructure for research collaboration in land use, transportation, and environmental planning." Journal of the Transportation Research Board No. 2163: 85–93, Transportation Research Record.

Flores-Cerrillo, J. (2010). "Autobox - State-of-the-art software for automatic time series analysis." Software Review, *OR/MS Today* (February).

Fourer, R. (2009). "Linear programming." Software Survey series, *OR/MS Today* 35, No. 3, June.

Hlupic, V. (1999). "Simulation software: Users' requirements." *Computers & Industrial Engineering* 37:185–188.

Karimi, H. A. (2009). *Handbook of research on geoinformatics.* Hershey and New York: IGI Global.

Lee, Y. C.; Zhang, G. Y. (1989). "Development of geographic information systems technology." *Journal of Surveying Engineering* 115, No. 3:304–323.

Maxwell, D. T. (2008). "Decision analysis: Find a tool that fits." Decision Analysis Software Survey Series, *OR/MS Today* 35, No. 5, October.

McCrary, S. W.; Benjamin, C. O.; Ambavanekar, V. E. (1996). "Consensus building model to select OASIS in small communities." *Journal of Urban Planning and Development* 122, No. 2:46–70.

Nash, S.G. (1998). "Software survey: Nonlinear programming" *OR/MS Today* 25, No. 3, June.

Ottensmann, J. R. (2000). "Applications of spreadsheet optimization capabilities in teaching planning methods: facility location and spatial interaction." *Journal of Planning Education and Research* 20, No. 2:247–258.

Ottensmann, J. R. (1985). *BASIC microcomputer programs for urban analysis and planning.* New York: Chapman and Hall.

Partyka, J.; Hall, R. (2010). "Vehicle routing software survey: On the road to connectivity." *OR/MS Today* 37, No.1, February.

Perry, M.; Sheth, A.; Arpinar, I. B.; Hakimpour, F. (2009). "Chapter XXI - Geospatial and Temporal Semantic Analytics." In *Handbook of research on geoinformatics*, edited by H. A. Karami. Hershey and New York: IGI Global.

Pintér, J. (2011). "Global optimization." From MathWorld—A Wolfram Web Resource.

Point of Beginning Magazine (2005). "2005 GIS software survey," June Issue. www.pobonline.com

Rey, S. J. (2009). "Show me the code: Spatial analysis and open source." *Journal of Geographical Systems* 11, No. 2:191–207.

Rey, S. J.; Anselin, L. (2006). "Recent advances in software for spatial analysis in the social sciences." *Geographical Analysis* 38:1–4.

Ritter, G; Wilson, J; Davidson, J. (1990) "Image algebra: An overview." *Computer Vision, Graphics and Image Processing* 49:297–331.

Rouch W (2007) "Second earth." *Technology Review* (July/August):39–48.

Star, J.; Estes, J. (1990). *Geographic information systems: An introduction*. Englewood Cliffs, New Jersey: Prentice Hall.

Steiniger, S.; Weibel, R. (2010). "GIS software." In *Encyclopedia of geography*, edited by B. Warf. London: Sage Pub.

Swain, J. J. (2007a). "Statistical software analysis survey: Finding a path in an uncertain world." *OR/MS Today*, 34, No. 1, February.

Swain, J. J. (2007b). "Discrete event simulation software: New frontiers in simulation," *OR/MS Today*, 34, No. 5, October.

Swain, J. J. (2009). "Discrete event simulation software: Boldly exploring new, old worlds." *OR/MS Today*, 36, No. 5, October.

Tomlin, C. (1990) *Geographic information systems and cartographic modeling*. Upper Saddle River, New Jersey: Prentice-Hall.

Vanderzee, D.; Singh, A. (1995). "Survey of geographical information system and image processing software." *International Journal of Remote Sensing*, 16, No. 2:383–389.

Wikipedia (2010). "Global optimization." Webpage page as last modified on 21 November 2010 at 16:12.

Wikipedia (2011) "Constraint programming," Webpage as last modified on 2 January 2011 at 12:23.

Woodward, W. A.; Elliott, A. C. (1983). "A survey of statistical packages on microcomputers." *Computational Statistics & Data Analysis* 1:191–200.

Yeung, A. K. W.; Hall, G. B. (2007). *Spatial database systems: Design, implementation and project management*. Dordrecht, Netherlands: Springer.

Yunes, T.; Aron, I. D.; Hooker, J. N. (2010). "An integrated solver for optimization problems." *Operations Research* 58, No. 2:342–356.

Yurkiewicz, J. (2010). "Forecasting: What can you predict for me?" Forecasting Software Survey. *OR/MS Today* (June).

8

A Software Survey of Analytics and Spatial Information Technology

"Get the facts first. You can distort them later."
Mark Twain

This chapter echos the discussion in Chapter 7, in which we laid out the landscape of analytics and its supporting software, with a special focus on spatial information technology. While the purpose of Chapter 7 is to provide the general picture, the current chapter will review solution methodologies and specific software packages. We start with a general-purpose applied mathematics software, such as MATLAB, and progress toward more specialized tools, ending with vehicle-routing software. Many commercial software packages are multi-purpose; they perform more than one function. Oracle Crystal Ball is a good example, being a key spreadsheet-based application suite for predictive modeling, forecasting, simulation, and optimization. The various packages by Vanguard Software are another example. Instead of listing a general purpose software several times, sometimes we chose to list it only once under what we judge to be the most appropriate use.

Much of the information is taken from the commercial software surveys available in the literature, particularly those published in *OR/MS Today*. This is supplemented with a heavy dose of public-domain software surveys gathered from a variety of sources. As with most surveys, the information is solicited from the vendors or the software developers, who may report their own product through "color glasses." Here, we try to do some judgmental screening. As a result, only a small subset of the products available in the market is listed here. Since we are aiming at a general audience, more "proven" and more robust software are favored over overly specialized software. These tend to be the more "popular" packages. Exceptions are made in the public-domain software, many of which are developed in research and academic environments, which by its very nature, is developmental and geared toward a particular niche.

While we include much more specifics in this review chapter than Chapter 7, the evaluation provided here is still quite general in nature, as we stay away from *endorsing* a particular software. The readers are strongly encouraged to visit the references for additional information that would help decide the best

Y. Chan, *Location Theory and Decision Analysis*, 2nd ed., DOI 10.1007/978-3-642-15663-2_8,

software for his or her application. Here, the author simply lists what he judges to be worth reporting. The criteria include the maturity and versatility of the commercial product, with attention duly placed on public-domain software as well. As always, the users should carefully judge for themselves the applicability of the product to his/her needs. In contrast to its functionality, the details of a software (such as its technical specifications and costs) are best gathered from the supplier's Website. Following our practice in this volume, no Web address is provided—inasmuch as a Web address is subject to change. This should not pose any inconvenience, however, as today's Internet search engine has greatly streamlined the location of Web addresses.

I. GENERAL ANALYTIC SOFTWARE

Echoing the layout in Chapter 7, we will divide software packages into two groups: general analytic software and spatial analytic software.

A. Spreadsheet Modeling

Spreadsheets are gaining popularity not only as a general office tool, but also as a modeling tool. The flexibility and versatility of a spreadsheet to solve a variety of problems are the driving force behind its popularity. There are a number of spreadsheet software available, including MS Excel and Lotus 1-2-3. It is not our intent to show our readers how to use a spreadsheet. Rather, we provide some guidelines to build *quality* spreadsheet models. Hillier and Lieberman (2009) suggest a four-step process to do so:

1. **Plan the spreadsheet model.** Visualize where one wants to finish and then do some calculations by hand to clarify the needed computations. Accordingly, a spreadsheet model is designed with the "feel and look" that one desires. The objective is a clear, logical layout to the overall model.
2. **Build the model.** It is advisable to start by building a small, readily manageable version of the model, mainly for experimentation.
3. **Test the model.** One is advised to test the small version first to get all the logic straightened out, before developing the full scale model.
4. **Analyze the model and its results.** In this final step, one applies the model to evaluate proposed solutions—to see how it performs. For more complex models, one may even use an add-in Solver to find the solution.

By itself, the four-step process is not sufficient. There are additional guidelines for building "good" spreadsheet models. Here are some salient ones (Hillier and Lieberman 2009, Winston and Albright 2007):

- □ Enter the data first, inasmuch as data really drives a spreadsheet.
- □ Organize and clearly identify the data.
- □ Enter each piece of data into one cell only.
- □ Separate data from formulas, instead of embedding data directly into a formula.

- □ Keep it simple by avoiding the use of powerful Excel functions when simpler functions are available that are easier to interpret.
- □ Use range names for easy reference, instead of just leaving the range in the cell.
- □ Use relative and absolute references to simplify copying formulas. This will save of re-entering the formula at various places in the spreadsheet.
- □ For readability, use cell comments liberally, use text boxes for assumptions and explanation, and use borders, shading, and colors to distinguish between cell types.
- □ Separate different parts of a model, possibly across multiple worksheets.
- □ When a Solver is used, the Solver uses a combination of the spreadsheet and the Solver dialogue box to specify the model to be solved. Therefore, it is possible to include certain elements of the model (such as the right-hand sides of the constraints in a linear program) in the Solver dialogue box without displaying them in the spreadsheet. For effective model dissemination, however, it is advisable to show the entire model on the spreadsheet.

To debug a spreadsheet model, check whether the output cells are giving correct results for various values of the changing cells. Also check whether range names refer to the appropriate cells, and whether formulas have been entered into output cells correctly. Toggle the worksheet between viewing the results in the output cells and the formulas entered into those output cells. Finally, other spreadsheet auditing tools can be used for additional debugging efforts.

Spreadsheets are appealing in part because it allows many models to be built without requiring computer programming skills on the part of the user. However, programming skills can be developed "on the job." The use of macros is an example. Beyond building a straightforward spreadsheet model, decision support systems (DSSs) can be developed that builds upon spreadsheets, when assisted by a bit of programming (Albright 2010). A computer language to facilitate this effort is **Visual Basic for Applications (VBA)**. For Microsoft users, VBA comes with MS Office, which make it accessible to a vast audience of potential users. To work properly, one needs to enable macros in the front, since VBA is really a macro to the Excel spreadsheet software. The end result of this effort is a set of spreadsheet applications with front ends and back ends, where inputs are enter in the front and where results are posted at the back.

For our discussions here, VBA works with Excel objects. A few typical Excel objects one would recognize include ranges, worksheets, workbooks, and charts. The goal is to expose a spreadsheet's object model and functionality to VBA, so that VBA can manipulate it programmatically. As a result, a DSS is built, going well beyond what a single spreadsheet can accomplish. In general, any application software package, such as Access, Word, or even a non-Microsoft software package, can similarly expose its object model and functionality to VBA.

B. Applied Mathematics

Aside from spreadsheets, applied mathematics packages offer a very general modeling environment for all sorts of problems. After all, the basic building block of a mathematical model is, well, mathematics.

1. MATLAB. The MathWorks now offers a very complex array of products to meet professional and scientific needs in both academia and industry. These products are organized around two main components, MATLAB and SIMULINK (Tarrazo 2006). Each of these products can be purchased separately, and a number of "toolboxes" and "blocksets" complement or extend each of them. The MATLAB "family" includes toolboxes in specialized areas and clusters of procedures. The former includes distributed computing, finance, bioinformatics, fuzzy logic, control, signal processing and communications. The latter include optimization, symbolic analysis, partial differential equations, genetic algorithms and direct search, statistics and data analysis, neural networks, splines, curve fitting, GARCH, wavelet, filter design, etc. The MATLAB family also includes a number of utilities, such as links for MS Excel, image acquisition, data acquisition, instrument control, datafeed, database, compiler, and report generator. Modeling needs are addressed by SIMULINK and related products.

Mathematical software is equally focused on both numerical and symbolic analysis. The symbolic toolboxes accompanying MATLAB provide access to MAPLE's analytical kernel, which has the capability to perform symbolic calculus, transforms, linear algebra, simplification of symbolic expressions, equation solving, specialized mathematical functions, general symbolic operations and variable precision arithmetic. The Extended Symbolic Toolbox also offers C code, Fortran and LaTex representation of symbolic expressions and full access to the most recent MAPLE kernel (except for graphics). This means support for programming in MAPLE and access to MAPLE specialized mathematics libraries.

According to Tarrazo (2006), the major difficulties stem from the following examples:

(a) Different versions of the program require different syntax, which limits the usefulness of the older codes. Also, the program itself is evolving. For example, current versions emphasize function handles and anonymous functions and discourage the use of inline functions.

(b) MATLAB syntax is sometimes hard to understand. Some procedures often seem to work well despite not following the recommended syntax, but this is without guarantees. The error messages sometimes refer to something different from what is going on. Some other items are hard to accustom to. For example, the anonymous function still looks like an odd construct after using it for a while. It is hard to explain the difference between the "abstract" and "symbolic" functions.

(c) The fact that MATLAB can be used in different ways is both an advantage and a source of perplexing errors. The remedy is to solve (at least at first) the same problem in different ways, until one knows what exactly the program will do.

2. OCTAVE. OCTAVE is a nice free alternative to MATLAB that permits users to process data or to use it as a general purpose (graphic) calculator (Le Reverend 2006). In some cases OCTAVE's syntax is slightly different from MATLAB's; but standard functions such as the creation of matrices, concatenation of matrices, two-dimensional and three-dimensional plots, data interpolation and numerical differentiation and integration are exactly the same. OCTAVE can be installed via

Fink or DarwinPorts, where Fink and DarwinPorts are the two major Unix porting alternatives for Mac OS X. Alternatively, the software can also be compiled from the available source code.

In OCTAVE, anyone can develop his/her own toolbox should the need arise. More important, s/he can then share their toolbox with the rest of the OCTAVE community—a prominent feature of an open-source code. An additional benefit of OCTAVE is that the community is very active and that it is very likely that the OCTAVE user group will help if one has problems using the software, from compilation of the source codes to the development of one's own libraries. In fact, the most successful projects are those not only with excellent code bases but thriving communities of users and developers. The social dynamics may lead toward new ways to organize science and heighten the pace of knowledge discovery.

Of course MATLAB, being a licensed commercial software, is more feature complete. The lack of a SIMULINK equivalent in OCTAVE is a problem for process control engineers. Since MATLAB 6.0, the software is no longer a "**Command Line Interface (CLI)**"-only tool. Some functions are accessible using a graphical user interface (GUI), including graphics and curve fitting. OCTAVE remains a pure CLI application and many value it as a good feature. Like MATLAB, OCTAVE is interpreted and can therefore be quite slow. If one tries to solve big problems, s/he can still use C++ routines directly in OCTAVE to help it run faster, but for small problems, one obtains the solution ten times faster with OCTAVE than the time spent developing the program in C++. OCTAVE also lacks a built-in editor but SCINTILLA will do the job.

Most of these limitations are not found in another MATLAB clone: SCILAB. SCILAB is being developed by a French consortium, but the syntax for SCILAB is quite different from MATLAB's. Therefore if one prefers to maintain as much compatibility as possible with MATLAB, OCTAVE is the more appropriate choice.

3. Mathematica. Mathematica has been known for its symbolic computation. On top of enhancements for symbolic computation and numerical computation, recent versions of Mathematica have gone well beyond these classic capabilities (Sodhi 2009). Included are advanced numerical analysis and linear programming, supporting interior point as well as simplex algorithms. Also included are sparse matrix manipulation with fast algorithms that compete with dedicated numerical software tools. The GUIKit allows a user to create a GUI running on top of Mathematica for other users to do specific tasks. Mathematica 5.1 gave access to Web services offered by other providers (e.g., Amazon) as a Mathematica function. It also allowed better access to spreadsheets (such as MS Excel) and databases.

Mathematica 6 would analyze notebooks files created by older versions to diagnose which function calls need to be modified. New features or enhancements include combinatorial optimization, constrained nonlinear optimization, exploratory data analysis, symbolic statistical computing and extended array operations. Of particular interest is the "manipulate" function, data visualization, and most of all, a variety of datasets including financial datasets. Instead of word processors such as Tex and LaTex, Mathematica features Publicon's ease-of-use in new versions.

Aside from graphic display and data manipulation extensions, there are features that take advantage of dual-core desktops that are very common today, not to say more specialized desktops (such as Apple) that have eight processors. Last but not least, the Wolfram Alpha search engine performs many search functions in Mathematica as other Internet engines. This shows that Mathematica has every indication and plan to provide desktop software that serves not only mathematics, but also day-to-day office applications.

C. Statistics

Following the criteria set up in Chapter 7 and using our best judgment, we screened the surveyed statistical software (Swain 2009, 2011) and resulted in the list shown below. What remains are some "substantive" statistical packages, including both specialized applications and full feature functionalities. We must admit that there may be bias toward software with full functionalities since we are looking for robustness and popularity. Each software is attributed to a vendor and reviewed with a brief commentary. Since data entry and output portability is a prime concern, we pay particular attention to the input formats and output formats of each software package. Although we exercise best judgment in the following screened list, the reader is advised to consult with the fuller listing by Swain (2009) for more details.

@RISK—Palisade Corporation
@ RISK performs risk analysis using Monte Carlo simulation to show many possible outcomes in Excel. Import formats include anything that can be brought into Excel. The application is an Excel add-in. Export formats include native Excel graphs and .jpg files. The new version, @RISK 5.5, introduced diverse new features and languages.

Autobox 6.0—Automatic Forecasting Systems, Inc.
Autobox will automatically build a customized model for univariate and multivariate time-series data—both in the batch mode and interactive mode. Import and export formats both include ASCII and Excel. The software can detect unusual behaviors such as level shifts, pulses, and seasonal pulses. Local time trends can be detected and adjusted automatically, accounting for intermittent demands while maintaining constancy of parameters and variance. The software was ranked the top Automated Software in the 2008 Daily Time Series Forecasting Competition.

DTREG—Phillip H. Sherrod
DTREG is a predictive modeling software that includes decision trees, neural nets, complete with vector machines support and genetic evolution functions. Import and export formats are in comma-separated values (.csv) files. Twelve predictive modeling methods are integrated into a unified program. The software is very easy to use.

JMP 8—SAS Institute
JMP 8 is an easy-to-use desktop solution, providing dynamic graphing, data visualization, comprehensive statistics and design of experiments. Import formats include JMP, Excel, text, Database via ODBC, SAS, html, First Choice Spreadsheet (FACS file extension), Access and dBASE. Export formats include JMP, Excel, text, SAS and dBASE. Notable features of JMP 8 are SAS integration, its capacity for choice experiments and modeling, and improved reliability/distribution fitting. Simulation can be performed within the design of experiments. JMP Pro Version 9 includes cross validation, Bootstrap Forests, Boosted Trees, 2-layer Neural Networks and more.

MaxDiff—Sawtooth Software, Inc.
MaxDiff provides "maximum difference scaling," or best/worst scaling of items such as product attributes, brand features, and position statements. Import and export formats are in .csv files.

Minitab 15—Minitab Inc.
Minitab is arguably a leading statistical software used for analysis and quality improvement worldwide, complete with powerful data and graphical analysis capabilities. Import formats are in Excel, XML, csv, txt, dat, qmd, and .dbf files of the dBase database management system; (Here, the .qmd file extension is primarily associated with the 'Quicken' software by Intuit Inc.). Export formats include Excel, XML, csv, txt, htm, html, rtf, etc.

Optimal Scientist Software Package—Transpower Corporation
The software helps to design and analyze optimal experiments. Import and export formats are in ASCII. The package determines the optimal number of predictor (input) variables and the resultant optimal regression equation. It also performs all-ways multiple regression.

SmartForecasts—Smart Software, Inc.
The software is designed for forecasting, sales/demand planning and inventory optimization. Import and export formats include text, spreadsheet files and database Star Schema, where the schema is the simplest style of data warehouse schema, consisting of possibly one "hub" fact table referencing any number of "peripheral" dimension tables—graphically depicted as a "star." SmartForecasts provides the Bootstrap methodology for intermittent demand, and full-holdout for time-series forecasting.

Smoothie—Demand Works
Smoothie is a sales and operations planning software for manufacturers and distributors, featuring Pivot Forecasting®. Import and export formats include Excel, text or database using ODBC. Smoothie's Pivot Forecasting enables immediate propagation of aggregate forecasts and adjustments at any level. Modules are now available for consensus demand planning, inventory policy simulations and analysis, and n-tier supply planning.

Stat::Fit—Geer Mountain Software Corp.
The software is popular among users who wish to statistically fit risk, simulation, and modeling distributions to user data. Version 2 includes 32 distributions and enhanced graphics. Import formats include ASCII. The software exports into specific formats for the simulation software of interest. Distribution viewer allows interactive display of distributions. Stat::Fit is complete with a derounding function and diverse data-manipulation options.

STATGRAPHICS—Centurion; STATGRAPHICS Web Services—StatPoint Technologies, Inc.
The powerful software is used for statistical data-analysis and modeling, quality control, design of experiments, forecasting and Six Sigma. Import formats are in Excel and .csv files, while exporting .csv files. StatAdvisor offers instant, easy-to-understand interpretations of one's statistical results. New features include Monte Carlo simulation, random generation of ARIMA time series, multivariate visualization, and sample size determination. The Web Services version is designed to be called from web applications. Web service returns results as html with imbedded images.

Unistat Statistical Package—Unistat Ltd.
Unistat is a comprehensive standalone package that can also work as an MS Excel add-in. Import and export formats include xls, wk!, csv, txt, sdf, slk, dif, mdb, dbf, html. Fully functional demo and/or the PDF manual can be downloaded from the software website.

XLMiner—statistics.com
XLMiner is a data mining add-in for MS Excel. It features classification, prediction, affinity analysis, data reduction, exploration and visualization. Import formats are in.csv files. Exporting data is possible despite the format or location of the data. Should the user choose to export directly, s/he must be willing to devote more resources to do so. XLMiner is equipped with a new time-series analysis, on top of the ability to save models for later review. Subsequently, XLMiner can score saved models to new data.

D. Simulation

A survey was conducted by Swain (2009) to collect information on both specialized and general simulation applications. There were about forty-odd products listed in the survey, taken from twenty odd vendors. This is one of the larger surveys on simulation software. The range and variety of these products continue to grow, reflecting the robustness of the products and the increasing sophistication of the users. The information elicited in the survey is intended to provide a general gauge of the product's capability, special features and usage. Our brief remarks below are meant to be introductory only, identifying the vendor and providing a brief commentary, based on the evaluation criteria set up in Chapter 7 of the text. For more details, the reader is referred to Swain (2009) and the vendor's Website.

@RISK—Palisade Corporation
For this software, RISKOptimizer combines genetic algorithm optimization with Monte Carlo simulation for optimization under uncertainty. @RISK provides a wide range of graphs, data, and statistics on simulation results. Input distribution fitting is based on Komogorov-Smirnov and Anderson-Darling goodness-of-fit tests, including cumulative and discrete distributions. Output analysis supports a wide range of graphs, data, and statistics on simulation results. @RISK Function Swap lets users remove @RISK functions from spreadsheet for non-@RISK use. Major new features include a new interface, new graphs, new functions, and much faster simulations. The version of @RISK, intended for industrial application, provides a fully customizable presentation and quality graphs. Input distribution fitting utilizes over 40 built-in distribution functions. Excel reports on the models can be shared with others who might lack the software to develop their own model. @RISK 5.5 has been fully translated into Spanish, German, French, Portuguese and Japanese.

AnyLogic—XJ Technologies
OptQuest by OpTek Systems Inc. provides optimization functionality to this simulation software. Input distribution fitting utilizes the Stat::Fit software. Simulation output provides dataset statistics, distributions, regular and two-dimensional histograms, various charts, etc. AnyLogic models can be exported as standalone Java applets or Java applications. Major new features include templates for agent based and other methods, the rail yard library, pedestrian dynamics modeling, and three-dimensional animation.

DecisionTools Suite—Palisade Corporation
The software features genetic algorithm for optimization under uncertainty, which is applied toward Monte Carlo simulation. Similar to another Palisade simulation package @RISK, input distribution fitting is based on Komolgorov-Smirnov and Anderson-Darling goodness-of-fits tests, including continuous and discrete distributions. The output includes a wide variety of graphs, data, and reports from simulation, utilizing decision trees and optimization analyses. @RISK Function Swap lets users remove @RISK functions from spreadsheet for non-@RISK use. Recent innovations include new graphs; new interfaces; new functions; faster simulations; and common interface conventions across Palisade products.

Emergency Department (ED) Simulator—ProModel Corporation
This medical-application software provides customized ED-specific data-output charts and graphs such as level-of-service, census by patient status and time-of-day and more. Input distribution fitting is based on user-defined distributions, or 15 predefined distributions, plus distribution fitting using Stat::Fit and/or Data Analyzer Software (at additional cost). Trial Version of Simulator can be shared. Solutions are driven by the ProModel VAO Technology, where VAO stands for Visualize, Analyze, Optimize. The software developer suggests that VAO can lead to better decisions faster.

Enterprise Portfolio—ProModel Corporation
Input distribution fitting is based on 15 predefined distributions. Distribution fitting can also be performed using Stat::Fit software (at additional cost). The software output analysis includes reports and charts, and documents in MS Excel format. A trial version of Portfolio Simulator can be shared among users. New products include web-browser version of the software, which runs on Microsoft (MS) Silverlight. It integrates with MS Project Server, including automatic updates. Solutions are driven by ProModel VAO Technology.

ExtendSim AT, ExtendSim OR, and ExtendSim Suite—Imagine That Inc.
An open-source evolutionary optimizer is included in all versions of ExtendSim. Input distribution fitting uses the Stat::Fit software, which is included in the package. For output, confidence intervals for output statistics are calculated at the click of a button. Free download Demo-Player version is available from the Imagine That! Website. The downloaded version opens, previews and runs the models of interest. Recent innovations include integrated database, built-in LP solver, revised and updated modeling components, variable connector arrays, and more scalable modeling. ExtendSim AT has broad functionality that supports modeling across both discrete-event and batch-process modes.

Flexsim and Flexsim CT—Flexsim Software Products, Inc.
For this simulation software, OptQuest by OpTek Systems Inc. provides the optimization functionality. Input distribution fitting is based on ExpertFit. Output is displayed as Flexsim Charts. Industry-specific and application-specific modeling objects and libraries or model-building objects can be shared. Flexsim Runtime allows completed models to be shared with others who might lack the software to develop their own model. Flexsim is easy to learn and it builds true three-dimensional models. New features include a complete library of model-building objects, consisting of container-terminal resources. Flexsim CT is the only commercial simulator designed specifically for managers and engineers to model container terminals. Meanwhile, Flexsim HC is a completely new simulation software product created specifically and solely to model healthcare patient-flow processes. Patient Trackú is the key to making healthcare modeling building both easy to do and extremely realistic and accurate.

GoldSim—GoldSim Technology Group
GoldSim includes a feature that provides global optimization of dynamic, uncertain systems, complete with sensitivity and uncertainty analysis. Model sharing is built into the software. New features include 64-bit support, new dashboard controls, enhanced reliability engineering and risk analysis, and enhanced distributed processing. A hybrid version combines system dynamics with aspects of discrete-event simulation, embedded within a Monte Carlo framework.

Micro Saint Sharp—Alion Science and Technology
The software can be linked with OptQuest optimization. Micro Saint Sharp automatically collects data to better understand the modeling process, in which data on utilization, queues, resources, and tasks are collected automatically. At the

same time, users can customize data collection to see whatever results are desired. For model sharing, users just need to select the Export Model to Runtime option under the Utilities menu. There s/he would select a folder, and Micro Saint Sharp will then create runtime version of the model that can be distributed. New features include three-dimensional animation, custom object types, communications module, VISIO import/export, and experiment definition.

Portfolio Simulator and Project Simulator—ProModel Corporation
Mathematical optimization capability is built into this stimulation tool. Input distribution fitting uses 15 predefined distributions, or the Stat::Fit software at additional cost. Output analysis includes reports and charts, with option in the MS Excel format. Trial Version of Portfolio Simulator can be shared among users. New features include direct imports from MS Project Server and Excel. Solutions are driven by ProModel's VAO Technology.

Process Simulator—ProModel Corporation
A notable feature of this simulation software is that output analysis reports and charts are included, with options in MS Excel and Access format for further analysis. In addition, model information can be modified in MS Excel and imported back into Process Simulator for additional runs. Model sharing can be accomplished through Process Simulator Lite. Recent additions include directly displaying simulated results via data graphics, and Minitab Integration. Solutions are driven by ProModel's VAO Technology.

ProModel Optimization Suite, ServiceModel Optimization Suite, MedModel In this software package, optimization Suite—ProModel Corporation
Optimization is available using OptQuest and/or SimRunner. Input distribution fitting uses user-defined distributions and 15 predefined distributions, plus distribution fitting using Stat::Fit that is included in the package. The software outputs analysis reports and charts, including documents in MS Excel and Access for further analysis. Model packaging is available within software view using free ProModel Play. The suites model separate areas of a broader model independently. Then it brings them together for overall simulation, complete with Minitab connectivity. Solutions are driven by ProModel's VAO Technology. The MedModel suite is a simulation-based software tool for evaluating, planning or re-designing healthcare systems.

Risk Solver, Risk Solver Platform, Risk Solver Premium—Frontline Systems Inc.
In this simulation package, input distribution fitting is available, matching against scores of continuous and discrete distributions. Risk Solver outputs charts, probability distribution function (PDF), cumulative density function (CDF), tornado and scatter plots, plus 30 statistics and risk measures. The Risk Solver Engine supports the sharing and distribution of models to others. New features include ultra-fast interactive simulation, probability management with **Statistically Improbable Phrases (SIPs)** or distributions, and multiple parameterized simulations. (Here, SIP is a search string likely to generate meaningful results from a search engine.) Notice the software is from developers of Excel Solver and Premium Solver, which are

among the notable products in spreadsheet solvers. Finally, Risk Solver is ungradable to Risk Solver Platform for powerful stochastic optimization. The Platform provides simulation and optimization, stochastic programming, and robust optimization with up to 12 powerful solvers. New features include everything in Risk Solver and Premium Solver Platform and more. The Premium software provides simulation and optimization with GRG Multistart[1] and Evolutionary Solvers. New features include everything in Risk Solver, everything in Premium Solver, plus Simulation Optimization.

Simcad Pro-Patented Dynamic Process Simulator—CreateASoft, Inc.
The software features built-in Dynamic Optimizer tool, on-the-fly user interaction, integrated work-order/schedule optimization, value stream maps, Gantt chart, scenario analysis and lean reports. Here, value stream mapping is a lean manufacturing technique used to analyze the flow of materials and information currently required to bring a product or service to a consumer. An input distribution is auto-fit to a database encoded in .csv file, and Excel files, etc. Model sharing can be accomplished through Simcad Viewer or Simcad Online. Recent advances include Multi-core Processor, Dynamic Optimizer, linkage with Radio Frequency Identification or Real-Time Locating System, Simcad Online, and Excel Import/Export Wizards.

SIMUL8 Standard, SIMUL8 Web, SIMUL8 Professional—SIMUL8 Corporation
This simulation software includes OptQuest optimization by OpTek Systems Inc. It provides automatic confidence interval calculation with no coding required. Input distribution fitting is accomplished through the Stat::Fit software. Outputs include results and charts for all simulation objects, dynamic on-screen reporting as the simulation executes, and export to external applications such as MS Excel, V.I.S.A, and Minitab. Here, V.I.S.A. is a Web based multi-criteria decision-making software. By linking SIMUL8 to V.I.S.A, one can assess the impact of weighing the importance of each of these performance measures, and assess which scenario best meets the analysis goals. SIMUL8 boasts being a pioneer on the use of trial calculators, which determines the number of simulation runs to get accurate confidence intervals. According to the software developer, SIMUL8 is easy to use, powerful, and among the fastest in the field. The web version allows hosting on the vendor's Website, user's Website or user's corporate network. There is an option for animation. No end-client install is required. The Professional version has all the features of the Standard version. On top, it has the SIMUL8 Results Manager, which provides centralized results database, scenario and run comparison reports, and customizable charting and reporting capabilities. Model sharing can be achieved through SIMUL8 Viewer. Among recent advances is a 30%-faster run execution speed, the SIMUL8 Results Manager, predictive text, multidimensional arrays, customizable runtime charts, and extended ease-of-use, and power to link to any application or data source with SQL and Component Object Model (COM).

Tecnomatix Plant Simulation—Siemens PLM Software
Tecnomatix Plant Simulation is a discrete-event simulation tool that creates digital models of logistic systems, so that one can explore the system's characteristics and

optimize its performance. The Integrated Optimizer features layout optimizer plus neural networks and bottleneck analyzer. Input distribution fitting is accomplished through the Data::fit module; Outputs include Sankey chart, html result report, and Gantt chart. A Sankey chart is a flow diagram, in which the width of the arrows is shown proportionally to the flow quantity. They are typically used to visualize material transfers between processes. Tecnomatix has integrated Pay and Go functionality, and fee-charge viewer for licences. For applications, Tecnomatix provides Virtual Commissioning, plant design and optimization solution, and Teamcenter Interface for product-lifecycle-management solution. Virtual commissioning is the use of a virtual model that represents an accurate and realistic three-dimensional simulation of mechanical, electrical, and control systems in order to validate the physical functions of a production system prior to actual physical implementation. In Tecnomatix, the user is provided with real object-orientation, inheritance[2], openness to import SAP, Excel, Oracle data, and ease-of-use through real Windows standards.

Vanguard Business Analytics Suite, Vanguard Strategic Forecasting Suite, Vanguard System—Vanguard Software
Features include simulation statistics, sensitivity analysis, and graphical presentations. Input distribution fitting is performed through the Distribution gallery, Auto-fit wizard, user-defined distributions, and SIP/SLURP search support. Here, SLURP is a web crawler from Yahoo! that obtains content for the Yahoo! Search engine. Outputs integrate with Microsoft Office, rendering them available as Web reports and Interactive Web-based models (without any Web programming). Modelers can take advantage of grid computing, collaborative modeling, and linkable models. Analytics Suite provides scalable, high performance simulations. Vanguard Strategic Forecasting Suite features a new stochastic optimizer and grid computing for higher performance. The Vanguard System is designed for large-scale enterprise modeling, including invariant branch optimization in compiler operations.[3] It includes Multi-Objective Decision Analysis (MODA)/MAUT, AHP, and decision tree analysis.

E. Optimization

While the review here is mainly on LP and MIP software, a number of the following products can handle more general nonlinear problems as well (Fourer 2009). Some of the available software packages for nonlinear programming are described in a survey by Nash (1998). Note that this latter survey was restricted to "full feature" nonlinear programming packages, meaning packages that accept a full range of non-linearities (i.e., nonlinear objective function, and nonlinear equality and inequality constraints). This omits many worthwhile pieces of software that only handle more restricted models.

For more specialized applications, there is a recent movement toward open-source optimization software. A prominent example is the COmputational INfrastructure for Operations Research, or COIN-OR for short. The project is managed by a non-profit foundation. Irrespective, the number of available commercial and public-domain optimization products is large. Accordingly, we have done a judgmental quality screening, resulting in the following short list.

ADBASE—University of Georgia

ADBASE is an MPS-based PC software that operates under DOS. It is built upon the revised simplex algorithm, as extended to multiple criteria. It generates all non-dominated multicriteria-LP solutions. Interval criterion weights can be specified to find a subset of the non-dominated solutions. Lexicographic ordering of criterion vectors is included as a feature in the software. An I-file is used to input the cost vectors for the criteria, the constraint matrix, and the right-hand-side vector. A G-file is used to specify the problem-specific options. ADBASE is among the very few software available to solve a multicriteria LP. While not totally user-friendly, it is a free software for the non-profit academic community. To obtain a copy of the executable code and a User's Manual, please contact Dr. Ralph Steuer at the University of Georgia.

AIMMS, the modeling system—Paragon Decision Technology Inc.

AIMMS is an integrated modeling tool built upon the use of large-scale optimization models. It is an integrated development, complete with end-user GUI, point-and-click database and XML integration, advanced developer-support tools, multi-language and unit support, internal data-management facilities, batch run options; multi-agent technology, API/COM interface and Web posting. Of particular interest is the outer approximation algorithm, an open source algorithm for generating the set of all efficient extreme points in the outcome set of a multicriteria LP. AIMMS provides a modeling environment for CPLEX, XPRESS, XA, CONOPT, KNITRO, LGO, BARON and more through their Open Solver Interface. New features include parallel solver sessions, stochastic programming support, new syntax editor, case differencing, web services, GIS support, pivot table, multi-developer support, non-linear math program inspector, solution pooling, lazy constraints,[4] Benders' decomposition algorithm, nonlinear presolve, multistart solve, GUROBI,[5] MOSEK,[6] Dynamic database functions, MS Virtual Earth link, Yahoo Maps, ESRI Shape files, geocoding functionality, free viewer license, and free student license.

AMPL—AMPL Optimization LLC

AMPL is a general nonlinear solver that supports second derivatives, detailed solver-specific directives and results, user-defined functions and MATLAB interfaces. Noted for its modeling environment, it supports at least 35 solvers, as listed on the vendor's Website. Flexible handling of sets and indexing for handling complex models naturally and large models efficiently. AMPL includes a scripting language for iterative optimization schemes. Free experimentation is available through the NEOS Server.[7] New features include Solver support for multiple solutions, parameter tuning, local search, mixed-integer programming with non-linearities.

BendX Stochastic Solver—Maximal Software, Inc.

BendX is a standard C Application-Programming-Interface (C-API) callable-library stochastic solver. It supports solving both scenario-based and independent-variable models with Deterministic Equivalent (DEQ) and Benders' algorithms.[8] BendX solves both DEQ and Benders decomposition problems, using CPLEX, GUROBI and CoinMP as the underlying LP solvers. In addition to the C-API callable library interface, BendX supports both SMPS[9] and XML files. With optimization projects, there is often a need to store model instances, e.g., for building model libraries, providing technical support, and optimization services over the Internet. OptML facilitates a new portable, non-solver specific standard, based on XML, which supports multiple problem types, including linear, mixed-integer, quadratic, nonlinear, and stochastic programs. At the same time, there is also a need to convert raw data in XML format into problem instances that conform to the optimization services instance language (OSiL) standard. BendX can be an add-in to the MPL Modeling System, CPLEX, GUROBI, and CoinMP. Recently, BendX offers unique object-oriented library stochastic interface for Visual Basic, C#, and Java.

CoinMP—Maximal Software, Inc.

CoinMP is an open source C-API interface library that includes Coin LP (CLP), Coin Branch-and-Cut (CBC), and Cut Generation Library (CGL) projects. Precompiled ready-to-use CoinMP.dll is available for download. When source is compiled for Windows it generates a CoinMP.dll library that can be readily used in projects. When compiled for Unix/Linux it generates a CoinMP.so library. CoinMP serves as an add-in to MPL Modeling System and others. New release of the software offers object-oriented library interfaces for Visual Basic, C#, and Java. Linux/Unix versions are available with automake/configure support.[10] IPOPT[11] and Storage Management Initiative support are coming soon.

GAMS—GAMS Development Corporation

GAMS is arguably a classic algebraic modeling language. A GAMS system includes all components with size restrictions removed for those that are purchased. Solvers/modeling environments that link to the product include ALPHAECP, BARON, CONOPT, CPLEX, DECIS, DICOPT, GUROBI, KNITRO, LGO, LINOGLOBAL, MINOS, MOSEK, MPSGE, MSNLP, OQNLP, OSL, PATH, SBB, SNOPT, XA, and XPRESS.

IBM ILOG CPLEX—ILOG, an IBM Company

ILOG CPLEX has been one of the front runners in solving LPs, exploiting the speed of network algorithms. With call backs and goals, users can customize MIP branch-and-cut, such as branching strategies and cutting planes. IBM ILOG ODM is an application and cutting-planes development tool. It builds and deploy custom planning and scheduling applications based on IBM ILOG CPLEX. OPL Development Studio has multi-model algorithms, featuring warm-start, external calls to Java, decision expressions, performance profiler, automatic tuning, constraint detection and conflict resolution. Solvers/modeling environments that link to the product include IBM ILOG OPL-CPLEX Development System, AIMMS, AMPL, GAMS, MPL, MATLAB, and Microsoft Solver Foundation.[12] New features include dynamic search, MIP solution pools, deterministic parallel MIP, tuning tool; multiple MIP starts, solution polishing API, multi-model algorithms, warm-start, external calls to Java, decision expressions, performance profiler, automatic tuning, constraint detection and conflict resolution.

LINDO API, LINGO—LINDO Systems, Inc.

LINDO API is a popular suite of fast callable solvers for creating customized linear, integer, nonlinear, quadratic, stochastic and global optimization applications. Solvers/modeling environments that link to the product include MATLAB, GAMS, LINGO and What'sBest. LINGO is a popular suite of fast linear, integer, nonlinear, quadratic, stochastic and global solvers. It includes a comprehensive modeling language with convenient data options. LINGO's solvers and interactive modeling environment make it a comprehensive tool for operations research professionals. The modeling language and mathematical functions allow quick, concise problem expression. Data can be stored separately in text. Solvers/modeling environments that link to the product include LINDO API and Excel. New features include Stochastic programming capabilities, statistical sampling, and K-best MIP solver.

MPL Modeling System, OptiMax Component Library—Maximal Software, Inc.
With the OptiMax Component Library which is listed separately below, MPL models can be embedded into end-user applications using Visual Basic, VBA, C#, C/C++, Java, and Web-scripting languages. OptiMax is an object-oriented component library, specifically designed to help embedding optimization models into end-user applications. Solvers/modeling environments that link to the product include CPLEX, GUROBI, Xpress, OSL, XA, MOPS, LINDO, FORTMP, C-WHIZ, CoinMP, GLPK, LPSOLVE, CONOPT, KNITRO, LGO, PATH, and EXCEL. Latest releases feature increased speed and scalability. New solver versions include CPLEX 12, GUROBI 1.1, Xpress 2008, MOPS, CoinMP, GLPK, LPSOLVE, KNITRO, CONOPT, LGO, PATH, stochastic programming, and new data sources. New release of OptiMax offers new language support, and more than 20 new objects have been added, with new enhanced methods and properties for advanced solver handling and data management.

Premium Solver Platform, Risk Solver Platform, Solver Platform SDK—Frontline Systems Inc.
Features of this software include convex and non-convex smooth nonlinear optimization, non-smooth optimization, global optimization and IF/MIN/MAX linearization. Premium Solver Platform can be an add-in to Microsoft EXCEL. There are five built-in solvers, eight plug-in solvers including LP/QP, GUROBI, Xpress-MP, MOSEK, KNITRO, LSGRG, LSSQP, and OptQuest. Recent release of the software includes new modeless user interface, parameterized optimization, charts/graphs, multi-core nonlinear and global solvers, and video demos. Compatible upgrade is obtainable from developers of Excel Solver and Premium Solver. Premium Solver Platform is upgradable to Risk Solver Platform for simulation and stochastic optimization. This powerful Excel Solver upgrade integrates conventional optimization, simulation/risk analysis, stochastic and robust optimization, and decision analysis. On top of the functions in Premium Solver, the Risk Solver Platform performs Monte Carlo simulation optimization, stochastic programming and robust optimization. The newly released product includes parameterized simulations, multi-core simulation, and decision trees. Working outside Excel, Solver Platform SDK has these additional features: object-oriented and procedural APIs for C/C++, C#, VB.NET, VB6/COM, Java, and MATLAB. New features include Visual Studio 2008 support, and a large library of examples. It now reads/writes MPS, LP, OSiL files; IntelliSense and JavaDocs.

SAS—SAS Institute Inc.
The comprehensive software suite includes, among other items, quadratic optimization using an interior point solver, general nonlinear optimization that includes nonlinear objective and/or nonlinear constraints. There are several techniques for general nonlinear optimization with boundary, general linear, and nonlinear constraints. Two algorithms are designed for quadratic optimization problems and two other algorithms address nonlinear least-squares problems.

SAS integrates optimization with data access and handling. Recent upgrade includes irreducible infeasible set analysis, enhancements to the OPTMODEL modeling language, and interior-point nonlinear-programming solver. Given its statistical background, SAS integrates optimization with data access and handling, statistical analysis and data mining, forecasting, reporting and deployment.

Smart Optimizer (SOPT) 4.2—SAITECH, Inc.
Various heuristic search algorithms are implemented in SOPT to look for integer feasible solutions quickly. Quadratic or smooth nonlinear problems are solved fast by an interior-point algorithm. Solvers/modeling environments that link to the product include AMPL. Extended search capabilities are further developed to find feasible solutions to large-scale integer programs. Cuts are automatically generated by user parameters.

SYMPHONY—COIN-OR Foundation
SYMPHONY is an open-source solver for mixed-integer linear programs written in C. It can be used either (1) as a callable library through either a modeling shell, or (2) as a standalone program. Features include an open-source solver for bi-objective integer programs, warm starting for integer programs, basic sensitivity analysis for integer programs, and call backs for customization. These custom modules are included for specific combinatorial problems: vehicle routing, set partitioning, multicriteria knapsack, network routing, etc. Solvers/modeling environments that link to the product include GMPL, AMPL, GAMS. Here, GMPL stands for GNU Mathematical Programming Language. GMPL is also referred to as GNU MathProg--the two terms being interchangeable. Both represent a high-level language for creating mathematical programming models.

Vanguard System for Web-based Optimization—Vanguard Software
This is a new tool to build and deploy Web-based optimization applications. It supports access controls, version controls, and systems integration. The recent release is a development tool for Web-based optimization, stochastic optimization, grid computing, Web services, and collaborative modeling. Other features include forecasting, Monte Carlo simulation, decision tree analysis, statistical analysis, financial analysis, and sensitivity analysis.

What'sBest—LINDO Systems, Inc.
What'sBest is a large-scale optimization add-in for MS Excel. It allows the user to build linear, nonlinear and integer models in one's favorite spreadsheet. It is powerful enough for real world models and ideal for building models for clients. Other solvers/modeling environments that link to the product include LINDO API. Recent release includes stochastic programming capabilities, statistical sampling, K-best MIP solver, and expanded function support.

XA—Sunset Software Technology
XA has been around for decades. Recent development experienced a five-times speedup in solving mixed integer linear programming models. XA can serve as add-in to EXTEND, EXCEL, PYTHON, and Goldsim. Other solvers/modeling environments that link to the product include AIMMS, GAMS, MPL, and AMPL. New features include Conflict Analysis, piecewise linear, and concurrent primal and dual algorithm.

F. Decision Analysis

Here are some decision analysis and multicriteria decision-making products to choose from. Most of them cover multi-objective decision analysis (MODA) and multi-attribute utility theory (MAUT). Again, we have done a judgmental screening, resulting in a selected list of more full-feature and popular packages. Once again, the reader is advised to consult additional references, particularly Buckshaw (2010) and Maxwell (2008). The Buckshaw and Maxwell surveys from the database from which we provide the following list. Whichever tool(s) are ultimately selected by our reader, they should be intuitive to the user, explainable to the client and support easy iterations among the various stages of the decision analytic process.

1000Minds—1000Minds Ltd.
The software includes MODA/MAUT. It also provides a procedure called PAPRIKA, which is based on the fundamental principle that any ranking of alternatives is uniquely defined by all possible pairwise comparisons of the alternatives—hence the acronym PAPRIKA, which stands for Potentially All Pairwise RanKings of all possible Alternatives. Another prominent building block is Conjoint Analysis, which involves surveying stakeholders about the relative importance of a product's (or service's) features. Recent release shows that the software now manages the entire process for developing a prioritization tool, including administrative functions for managing participants. The software features value for money analysis; selection of portfolio of alternatives with budget constraints; analysis of group elicitation, prioritization of patients for health care, selecting health technologies, project portfolio selection, and competition judging.

Crystal Ball Standard, Professional & Premium Editions—Oracle
The generalized software package includes, among other analytical features, MODA/MAUT. Other algorithms implemented package include Monte Carlo, linear and nonlinear programming. The 2008 release includes a new version of OptQuest. The new version features a new Wizard for setting up optimization procedures; full integration with Excel and Crystal Ball, including the ability to control optimization through Crystal Ball's control panel, and an updated version of OptQuest's global optimization engine. The new release supports both linear and nonlinear constraints. It includes a more aggressive algorithm. It caters for new variable types, including binary, category, and custom. It has the ability to create reports and extract data. And it includes a Developer's Kit of API for programming optimization functions.

DPL—Syncopation Software
Features in DPL include MODA/MAUT, decision tree roll back, and Monte Carlo. New features include Developer API, initial decision alternatives tornado, default states, arrays in the influence diagram, and database INIT links. Marketed under DPL Professional, DPL Enterprise, and DPL Portfolio, of particular interest is the capacity to control DPL from another application using OLE Automation,[13] ability to build a custom front-end interface and invoke DPL in the background, and the aggregation of expert assessments. According to the vendor, DPL combines decision trees, influence diagrams and Excel spreadsheets to provide an intuitive and comprehensive modeling environment. Applications include portfolio prioritization, strategy, capital budgeting, and valuation.

ForeTellÆ—DecisionPath, Inc.
ForeTellÆ provides hybrid synthesis of Monte Carlo, system dynamics, agents/game theory, process and event models. There are automated RAD game tools[14] to build models from enterprise database and loading simulation outputs back into repositories. ForeTell enables a user to "test drive" critical organizational decisions before committing to implement them. Recent features include bi-directional interfaces to BI Solutions on RDBMs and data warehouses, plus bar chart analytics.

Hiview3—Catalyze Ltd.
Hiview3 features include MODA/MAUT. Recent release provides model templates, the ability to deal with unknown data, improved reporting, and analytical and aesthetic improvement. A network version is also available. Extended features include model sharing. The user can fix the entire model and allow others to use Hiview3 to explore the model as a live document, or fix the structure and scores and invite others to update the weights to see how their judgements influence the results. Applications include evaluating capital projects, analyzing policy settings, strategy selection, relocation/site selection, and budget resourcing.

RPM-Decisions—Systems Analysis Laboratory, Helsinki University of Technology
RPM stands for Robust Portfolio Modeling. The software is built upon Multiattribute Value Theory, considering incomplete information, non-dominated portfolios and core indexes. RPM defines a project's core index as the share of non-dominated portfolios that include the project. Recent extensions include project interdependencies, incomplete cost information and variable budget levels. The problem is formulated as a multi-objective zero-one linear programming problem with interval-valued objective function coefficients.

SMILE (Structural Modeling, Inference, and Learning Engine)—University of Pittsburgh
SMILE is a fully portable library of C++ classes implementing decision-theoretic methods, such as Bayesian networks and influence diagrams. Its Windows user interface, GeNIe, is a versatile and user-friendly development environment for graphical implementation. Both modules—Bayesian networks and influence diagrams—have been made available to the community free-of-charge since 1998 and have now several thousand users worldwide. Contact Dr. Marek J. Druzdzel at the Decision Systems Laboratory, University of Pittsburgh, if interested.

II. SPATIAL ANALYTICS SOFTWARE

In Chapter 7, we discussed the salient features of spatial analytic software, including GIS, image processing, and vehicle routing. Here, we will provide a list of software that subscribe to these features.

A. GIS

There are many GIS software packages offered, but only a handful is true benchmarks (Galati 2006). Notable ones on the commercial market include ESRI's ArcGIS, Intergraph Corporation's G/Technology, General Electric's Smallworld, Clark Labs' IDRISI Kilimanjaro, Autodesk's GIS Design Server, Delorme's Xmap, and Pitney Bowes' MapInfo. Most are developed with full GIS functionality, concomitant with a corresponding price tag. Perhaps worth mentioning is Delorme's Xmap and IDRISI Kilimanjaro, which are lower cost options among this peer group, with the latter characterized by user-friendliness and built around raster-based instead of vector-based files.

Unlike the generalized analytics software listed above, GISs are fewer in number, as they are specialized packages for spatial application only. There are currently only a handful of GIS packages, but many more are expected to be published in the next decades (Prastacos 1992). For the packages listed below, they are constantly being improved.

ARC/GIS—ESRI
This is the most widely-used GIS system available for a variety of computers, including desktop, laptop, tablets, servers and mobile devices. It is a powerful, command-driven GIS with extensive capabilities for data storage, editing, display, and geographic analysis. Users can install plug-in's, called extensions, to add functionality. ArcExplorer is a free GIS data viewer that allows basic mapping and spatial querying. ARC/GIS has been a leader in the GIS software market.

GIS Design Server—Autodesk
This package is from a company that developed AutoCAD, a computer-aided-design tool familiar to the engineering community. Through AutoCAD, Autodesk has earned user trust in the community. For GIS, what the developer offers is Autodesk Map, which designs, maintains and produces maps and geographic data. The program suite supports desktop, laptop, tablet, and mobile platforms. On top of this, the GIS Design Server provides an environment that allows flexible data integration, although the integration is most seamless with AutoCAD files. With the familiar program exchange through AutoCAD, Autodesk's GIS product commands a huge following.

G/Technology and GeoMedia Software Suite—Intergraph Corporation
The G/Technology suite of programs offers industry-specific data models for utility, pipeline, water and communications companies. Its open GIS architecture allows it to work with many GIS formats. The companion GeoMedia Software Suite facilitates map design, presentation and sharing. The software allows desktop, laptop and enterprise wide compatibility. GeoMedia viewer is a free GIS viewer, facilitating desktop geospatial viewing and the sharing of data among users. The combination of G/Technology and GeoMedia offers a formidable GIS environment.

IDRISI Kilimanjaro—Clark Labs
With an academic genesis, IDRISI Kilimanjaro is a widely used raster based GIS and image processing software. It is user friendly and highly accessible. Over the years, it has become a benchmark in geospatial standard. Its functionality includes modeling, database querying, spatial data development and geostatistics. Its low cost and research functionality explain why its object-oriented development tools

are popular for focused research. IDRISI Kilimanjaro has been particularly popular for environmental analysis and modeling.

MapInfo—Pitney Bowes Business Insight
MapInfo is a menu-based, user-friendly desktop mapping and GIS systems that can store and display street networks and zone boundaries. It has sophisticated routines for geocoding. The proprietary data structure is not topological, hence paths and routes, for example, cannot be defined. MapInfo is noted for its capacity to allow further development. For example, MapX is an Active X component that enables active software embedding, allowing embedding mapping applications within other applications, such as MS Word, Excel, and Lotus 123.

Smallworld Suite—General Electric
A unique feature of the Smallworld Suite is its advanced spatial technology and seamless existing system integration. Key components include the Core Spatial Technology, Spatial Intelligence, Enterprise Integration Tools, and Design Manager. The software offers desktop, laptop and Internet interoperability. Its architecture is different from ESRI's all purpose design, and Intergraph's specialized and all-purpose programs. Unlike these broadly focused programs, the Smallworld Suite centers upon engineering, scientific and business-oriented applications.

TransCAD—Caliper Corporation
TransCAD is a powerful and easy-to-use GIS-based transportation package. The system consists of two parts: a GIS engine and a tool box of transportation models and procedures. The GIS engine is menu driven and, in addition to the standard GIS functions, can directly support transportation data structures such as nodes, links, networks, paths, and tours. TransCAD has a set of dynamic segmentation and linear referencing tools for managing highway, rail, pipeline, and other networks. TransCAD also provides a platform for users to develop their own transportation-related models. The software is developed for the desktop PC MS operating systems, including Windows 2000, Windows XP, Windows Vista, and Windows 7.

Xmap—Delorme
Through Xmap, the mapping giant Delorme offers a low-cost GIS and GPS mapping capability, complete with robust functionality. Xmap's modular design allows for expandability and interoperability. XMap is a three-tiered GIS software suite designed for transfer of information between GIS administrators and field personnel. XMap Enterprise provides database management tools and is intended for corporate GIS administration and data deployment. XMap Editor, a full featured GIS, offers an extensive set of GIS layer importing, creating and editing tools, ideally suited for small scale GIS operations. XMap Professional is primarily a GIS data viewing application. However, when used in conjunction with XMap Enterprise, it becomes a proficient field data collection and updating tool, ideally suited for field personnel. Xmap supports open GIS and most GIS data formats.

Prominent among the public domain software is Geographic Resources Analysis Support System (GRASS), a fully functional GIS environment. Originally developed by the U.S. Army Construction Engineering Research Laboratories, it has been maintained by Baylor University in Waco, Texas since 1995. Other free GIS software has been developed by university researchers, programmers, philanthropists, geospatial organizations, governmental agencies, and private developers since the 1980's. Listings of these freeware can be found on the Web through FreeGIS.org, GIS Lounge and the Open Geospatial Consortium. A parallel organization, OpenGIS, is dedicated to developing and standardizing geospatial and geo-processing specifications. Open Source programs are applications of which one can access the source code. Listed here are available open-source GIS-based applications one can download, written for a variety of platforms and in various languages.

GRASS—Baylor University
GRASS is a public-domain raster GIS, a vector GIS, an image-processing system, and a graphics-production system. It is extensively used at government offices, universities, and commercial organizations. It is written mostly in C for Unix. GRASS is a powerful but often difficult to use GIS program, being a command-line software. Quantum GIS is currently implementing an easier interface for GRASS's capabilities. Meanwhile, a Java version of GRASS (JGRASS) is being built on top of uDig, where uDig was built and maintained by HydroloGIS—concentrating on hydrogeological and geomorphological capabilities. Rather than duplicating the effort of uDig and GeoTools, the JGrass team chose to focus on the unique parts of their project, which are tools and algorithms. At the same time, the team gets the basic infrastructure—consisting of vectors, formats, re-projection, and workbench—from the uDig framework.

Quantum GIS—QGIS Development Team
Quantum GIS (often abbreviated to QGIS) is a free desktop GIS application that provides data viewing, editing, and analysis capabilities. QGIS runs on Linux, Unix, Mac OSX, and Windows. Quantum GIS is written in C++, and its GUI uses the Qt library. Quantum GIS allows integration of plug-in's developed using either C++ or Python. Qt library provides the cross-platform application development framework. Supported by other software, QGIS provides integration with other open-source GIS packages, including PostGIS, GRASS, and MapServer to give users extensive functionality. QGIS is continually maintained by an active group of volunteer developers who regularly release updates and bug fixes.

A GIS-component software is a building block that, when added to GIS software, forms an enhanced, personalized environment for the user. A specific-function component performs a dedicated task that adds to the GIS-environment tools. Such components include data format converters, flow-data analyzers, and image-processing software. User-development software, on the other hand, is a development toolkit that enables the user to program components to perform specific functions. For example, one may need to embed maps into a non-GIS program, and there is no pre-developed component for a raster-only GIS. User-development software is the only answer in this case.

A very popular component software is Geotools, an open-source GIS development toolkit that is freely distributed. The software is Java based and offers users the ability to develop Open-GIS-compliant products. What makes Geotools attractive to users is its modular design, which allows easy installation and removal of components. The software works well with Java fee-based and free GIS environments (Bruce 2007).

GeoTools—Open Source Geospatial Foundation (OSGeo)
GeoTools is an open-source Java-code library which provides standards compliant methods for the manipulation of geospatial data, for example, to implement GIS. Specifically, it is distributed under the GNU Lesser General Public License (LGPL). The GeoTools library implements Open Geospatial Consortium (OGC) specifications as they are developed. Geotools is used by a number of projects including Web Feature Servers, Web Map Servers, and desktop applications. GeoTools' modular architecture allows extra functionality to be easily incorporated. GeoTools aims to support existing-or-evolving OpenGIS and other relevant standards.

MapServer—OSGeo
MapServer is an open-source development environment for building spatially-enabled Internet applications. It can run as a **Common Gateway Interface (CGI)** program or via Mapscript which supports several programming languages. Here, CGI is a standard that defines how Webserver software can delegate the generation of Webpages to a console application. MapServer renders data for spatially-enabled Internet applications. It has excellent cartographic output. It can be used both as a WMS and WFS server and client. Here, WMS stands for Web Map Service—a standard protocol for serving georeferenced map images over the Internet that are generated by a map server using data from a GIS database. And WFS stands for Web Feature Service interface standard. WFS provides an interface allowing requests for geographical features across the Web using platform-independent calls. It can deal with a large amount of vector and raster data formats. It supports many scripting languages for developing Internet applications, e.g., PHP, Python, C, C++, C#, Perl, Ruby, and Java. Other functionalities include on-the-fly map projection.

Finally, previously created databases and bi-product datasets are distributed primarily through geospatial data clearinghouses, data warehouses, and data depots. The reader is referred to Data Depot, which is dedicated to free data and metadata. For a listing of data sites, refer to the University of Edinburgh - Association of Geographic information's GIS Resource list for links to several hundred U.S. and international GIS data sites.

B. Image Processing

In a survey, Vanderzee and Singh (1995) found out that there was not a direct relationship between functional capability and price for commercial image-processing software, where the products ranged in price from a few hundred to several tens of thousands of dollars. In the area of full-featured image processing (IP) systems, ERDAS Inc. was the leader. And for both basic GIS and IP capability at a low price, IDRIST's product for PC's has been the leader. For AM/FM, Accugraph has been the leader.

There were also some capable systems in the public domain, to which we will devote the bulk of the discussions here. While a few of these programs are general image analysis/manipulation programs, most are specifically designed to display and analyze satellite or aerial imagery (Pawlowicz 2009). In the following, we will group applications alphabetically rather than by function. Notice there are a number of general purpose GIS programs that also include significant satellite/aerial imagery functionality. They have been covered above under the GIS review section, and will not be repeated here.

FWTools—Frank Warmerdam

FWTools is a set of open-source GIS binaries for Windows (win32) and Linux (x86 32bit) systems produced by Frank Warmerdam (i.e., FW). The kits are intended to be easy for end users to install and deploy. There is no need to build from source, or having to collect lots of interrelated packages. FWTools includes OpenEV, GDAL, MapServer, PROJ.4 and OGDI as well as some supporting components. OpenEV is an open source library and reference application for viewing and analyzing raster and vector geospatial data. GDAL stands for Geospatial Data Abstraction Library, and is a veritable tool set of GIS data functionality. As reviewed in the GIS section, MapServer is an Open Source platform for publishing spatial data and interactive mapping applications to the Web. PROJ.4-Cartographic Projections Library is a GIS package that offers command-line tools and a library for performing respective forward and inverse transformation of cartographic data to or from Cartesian data with a wide range of selectable projection functions. Overall, the FWTools kit aims to include the latest development versions of the packages as opposed to official releases.

GVAR—Dartcom

GVAR stands for GOES VARiable format image acquisition, display and processing system. Dartcom supports GVAR data from GOES 8, 9, 10, 11, 12 and 13 with automatic detection during ingest. The system acquires high-resolution digital data (0.8 km visible, 4 km infrared) with calibrated temperature read-outs from infrared images. The system comes with fully automatic Windows-based GVAR Ingester, a data-ingest software. A companion software MacroPro automatically processes the acquired data to enhance, mask, print, animate, re-project and create products. A third software iDAP further displays and processes the data for image enhancement, product creation, projection transformation, land and sea masking, printing and exporting. As such, GVAR is among a handful of tools for aviation weather information, storm warning systems, forecasting, agriculture, oceanographic studies, and environmental and meteorological programs.

HEG—NASA

Hierarchical Data Format (HDF) is the prescribed format for standard data products that are derived from Earth Observing System (EOS) missions. HDF-EOS is a self-describing file format for transfer of various types of data between different machines based upon HDF. HDF-EOS is a standard format to store data collected from EOS satellites such as TERRA, AQUA and AURA. GeoTIFF is a GIS compatible format under a public-domain metadata standard which allows geo-referencing information to be embedded within a TIFF file. HEG stands for HDF-EOS to GIS, and is the acronym of a data converter. The HDF-EOS to GeoTIFF conversion tool (HEG) is developed to allow a user to reformat, re-project and perform stitching/mosaicing and subsetting operations on HDF-EOS objects. The output GeoTIFF file is ingestible into commonly used GIS applications. HEG will also write to HDF-EOS Grid and SWATH formats (i.e., for subsetting purposes) and native (or raw) binary. HEG presently works with MODIS (AQUA and TERRA), ASTER, MISR, AIRS, and AMSR-E HDF-EOS datasets.

HighView—various sources

The free trial version of GUI-based HighView is fully functional for band combination of 8- or 16-bit satellite imagery. These images include the global orthorectified

Landsat 7 ETM+ imagery available at USGS GloVis and the Global Land Cover Facility. Also included is the Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER), which is a high resolution imaging instrument that is flying on the TERRA satellite. Other satellites include SPOT, QuickBird and IKONOS. There is no limitation on image size and output format. Stretched output in 24-bit BMP format and/or un-stretched output in native GeoTIFF format can be readily used as base maps or backdrops in major GIS software, such as MapInfo and ArcGIS. Various options of linear and nonlinear stretches are allowed during the band combination.

NASA HDF-EOS Web GIS Software Suite (NWGISS)—NASA
NWGISS is a suite of web GIS software that makes HDF-EOS data available to GIS users based on Open GIS Consortium's (OGC) interoperability protocols. It consists of the following components: a map server (WMS), a coverage server (WCS), a catalog server, a Multi-Protocol Geoinformation Client (MPGC), and a toolbox. Those components can work both independently or collaboratively. The toolbox consists of two-way translators between HDF-EOS and major GIS formats, as well as the CreateCapabilities tool that automatically creates the XML capabilities descriptions from the metadata in HDF-EOS files. Both tools are available now. NWGISS map and coverage servers have been used by NASA and other space agencies. NWGISS is free to data providers who want to serve HDF-EOS data to GIS clients.

SamplePoint—NASA Goddard Space Flight Center
SamplePoint is a manual image-analysis program designed to facilitate vegetation cover measurements from nadir digital images of any scale. Here nadir is a point on the celestial sphere directly below the observer, referring to the downward-facing viewing geometry of an orbiting satellite. Operating essentially as a digital point frame, the software loads images, places classification points on the image, and stores classification data to a database as the user classifies each point. Up to three simultaneous views of each classification point, at varying zoom levels, are possible. The software appears to be primarily for close-up vegetation-cover analysis, but may be useful for other applications as well. Installation file contains SamplePoint, SPTracker, a Help Manual, a PowerPoint Tutorial and two sample images. The software is recommended for calibrating the threshold-detection level of image-analysis software or for making direct measurements of percent occurrence from digital images.

StarSpan—University of California at Davis
StarSpan is designed to bridge the raster and vector worlds of spatial analysis using fast algorithms for pixel-level extraction from geometry features such as points, lines, polygons. StarSpan generates databases of extracted pixel values from one or a set of raster images, and fuse them with the ancillary database attributes from the vector files. This allows a user to do statistical analysis of the pixel and attribute data in many existing packages and can greatly speed up classification training and testing. This feature is also found in other "mainstream" GIS software, such as ArcGIS and GRASS. However Booth et al. (2006) found that these two "mainstream" software have their limitations and neither really handles categorical raster summaries by polygon. They found that StarSpan appeared to be a more efficient option in terms of speed, scriptability and capabilities.

TerraLook—U.S. Geological Survey
TerraLook is a collaborative project that provides access to satellite images for users that lack prior experience with remote-sensing or GIS technology. The TerraLook Archive contains thousands of satellite images from the TERRA and LANDSAT satellites. Formerly known as the Protected Area Archive, TerraLook combines collections of geo-referenced JPEG images with a set of simple visualization and analysis tools. This allows users to explore the data and employ it for useful purposes in a variety of disciplines including conservation, development planning, education, urban studies, disaster planning and response, and others. It may be of particular use in developing countries that may have less capacity to purchase or work with remote sensing data. TerraLook is built on top of OpenEV, where OpenEV is an open-source software library and application for viewing and analyzing raster and vector geospatial data.

C. Routing

Routing software has come a long way since providing simply a route and schedule (Partyka and Hall 2010). Many of the following reviewed packages have more comprehensive functions for data keeping, analysis and planning. There is a clear trend to link routing software with tracking functions through GPS devices. The ability to communicate with mobile computing platforms is gaining prominence as computers are getting more portable, and as the tracking function becomes more prominent. Again, the list below represents a selected number of software that has been screened by the author. The screened packages tend to be software with broader functionality, and they have been on the market long enough to attract a clientele. Readers are encouraged to go to the references for more details when they wish to consider acquiring a particular software.

Accellos One Optimize—Prophesy Transportation Solutions, an Accellos Company
Integrated with Maptuit, Accellos One Optimize is a two-way-connected routing technology. Drivers receive real-time, turn-by-turn driving directions. Prophesy has implemented a new proprietary integration module for quick and common integration with other related software. Accellos One Optimize can communicate with cell phones, black boxes, and various Mobile Data Terminals. Currently, Accellos One Optimize is shaping up to be a full suite of supply-chain execution software and solutions for the industry. Clients include Boston Beer, Gold Medal Bakery, and Piggly Wiggly.

Paragon Routing and Scheduling Optimizer—Paragon Software Systems, Inc.
The software provides single/multi-site/integrated fleets planning. It is linked with truck tracking, resulting in actual movements tracked against the schedule. Paragon can be fully linked with satellite navigation and proof-of-delivery technology. Paragon's multi-tripping function optimizes resource in double dispatch operations.[15] Clients of Paragon include Airgas, McLane Company, CEVA, Exel Logistics, Toyota Material Handling, National Food Corporation, Red Ball Oxygen, and Ryder.

Roadnet Anywhere, Roadnet Transportation Suite—UPS Logistics Technologies
Roadnet Anywhere is a Web-based, easy-to-use daily routing and GPS tracking application. Through a Web-enabled application, Roadnet Anywhere captures

vital historical data that the user can review at any point, including historical traffic of "breadcrumb trails" on completed routes. The software also communicates with mobile devices. Altogether, the technologies enable a paperless office, GPS tracking, and pro-active service failure detection. Roadnet Transportation Suite performs strategic planning and analysis of daily route operations. The software records historical traffic, commercial road restrictions, and can perform CO_2-emission calculation. It includes multiple Web-based reporting tools for daily and historical analysis. Roadnet clients include Otis Spunkmeyer, Goodness Greeness, Lion Plumbing, and Oxygen One. Clients of Roadnet Transportation Suite include Anheuser-Busch, Office Depot, Sysco, Mohawk Industries, and Apria Healthcare.

StreetSync Basic, StreetSync Desktop—RouteSolutions

StreetSync Basic is a Web-based subscription routing-system. It allows integration with commercial Garmin and TomTom GPS devices. The one-click import and export function allows import from Excel or Access, and export to Garmin and TomTom units. Advanced integration with TomTom WORK is also possible, providing fleet-management and fleet-tracking solution combined with GPS navigation. Meanwhile, a distinguishing feature of StreetSync Desktop is an integrated customer database for analysis and planning. Clients of StreetSync Basic include Walco International Incorporated. Clients of StreetSync Desktop include Navteq, Coca-Cola Enterprises, Cintas, and Duncan Telecom.

III. CONCLUDING COMMENTS

The IT community is moving to tools like extensible markup language (XML), service oriented architectures (SOA), and Web services that facilitate distributed computing. XML's design goals emphasize simplicity, generality, and usability over the Internet. It is a textual data format with strong support via Unicode for the languages of the world. XML, SOA, and Web services have facilitated the growing prevalence of software as a service; that is, software residing on a server that is accessed by numerous client machines over a network, as opposed to software residing in multiple copies on its users' machines.

Sometimes referred to as "cloud computing," this new movement requires a set of standards (or protocols) when adopted for a particular application. Let us use a classic analytic tool such as optimization as an example. Fourer et al. (2010) are designing a platform called Optimization Service (OS) to implement cloud computing for optimization. The OS standards or protocols in this case include

- □ registration and discovery of optimization-related services in a distributed environment;
- □ representation of optimization instances, results, and solver options; and
- □ communication between a client on the user's end and solvers.

LogicBlox (http://www.logicblox.com), developer of online predictive and optimization software, is currently developing a product based on OS. This product allows users to develop optimization models through a Web-based graphical user interface. A model instance is sent to a solver on a local or remote machine; the underlying result is returned, where it is then converted into a more user-friendly solution report. A browser is the only required software on the client.

Following the analytics schema laid out in Chapter 7, the current chapter specifically reviews solution methodologies and software configurations. Once again, we like to conclude by suggesting that the software listed here are screened subject to our best judgment. We tend to favor more "popular" packages over the more obscure ones. For this reason, its inclusion does not imply our endorsement of the product. What shows up in this exercise is that there appears to be a trend to consolidate commercial software into comprehensive packages that perform multiple functions. These packages tend to be more popular and hence command a larger market share. Examples of these consolidated packages include Crystal Ball, Vanguard System and more.

In Chapter 7 and here, we make a distinction between general vs. spatial analytic software. As alluded to earlier, the fine line between general and spatial-analytic software is not as distinct as it used to be. AIMMS, the modeling system, is listed under optimization, but it has the following spatial-analytic functions: GIS, MS Virtual Earth link, Yahoo Maps, ESRI Shape files, and geocoding functionality. Thus the review here highlights a point that we made in Chapter 7: There is an emerging market for spatial information technology (IT), as evidenced by the increasing number of commercial routing, GIS and image-processing software. At the same time, we stipulate that while the demand is growing, the market is not strong enough to support some rather specialized applications, such as facility-location models, spatial statistics software and to a lesser degree image processing. This explains why we reported no commercial facility-location software here in this chapter. Instead, there are quite a few open-source or free spatial-IT products. These products may find their niche in the commercial marketplace in the future as demand grows over time.

ENDNOTES

[1] Most optimization software employs the generalized reduced gradient (GRG) methods for global optimization. However, multistart methods can overcome some of the limitations of the GRG Solving method alone. The multistart methods will automatically run the GRG method from a number of starting points and will display the best of several locally optimal solutions found. Because the starting points are selected at random and then "clustered" together, they will provide a reasonable degree of "coverage" of the space enclosed by the bounds on the variables. As a result, it is highly probable that the best local optimum is the global optimum.

[2] Inheritance is the process by which new classes called derived classes are created from existing classes called base classes. The derived classes have all the features of the base class and the -programmer can choose to add new features specific to the newly created derived class. According to Wikipedia, inheritance is what separates abstract-data-type (ADT) programming from object-oriented programming. ADTs are often implemented as modules: the module's interface declares procedures that correspond to the ADT operations, sometimes with comments that describe the constraints. This information-hiding strategy allows the implementation of the module to be changed without disturbing the client programs. The notion of ADTs is related to the concept of data abstraction, which is important in object-oriented programming.

[3] Optimization procedures in a traditional compiler are applied sequentially, with each optimization operation destructively modifying the program to produce a transformed program. The transformed program is then passed to the next optimization. Incremental computation of this kind takes advantage of repeated computations on inputs that differ slightly from one another, computing each output efficiently by exploiting the previous output. Since every non-trivial computation proceeds by

recursion, the approach can be used for achieving efficient computation in general. The key is to compute each iteration incrementally using an appropriate program. Tate et al. (2009) presented such an approach for structuring the optimization phase of a compiler. In their approach, optimizations take the form of *equality analyses* that add equality information to a common intermediate representation. Iterative program transformation is accomplished by direct tree manipulation. The Tate et al. optimizer works by repeatedly applying these analyses to infer equivalences between program fragments, thus saturating the intermediate representation with equalities. Once saturated, the intermediate representation encodes multiple optimized versions of the input program. At this point, a profitability heuristic describes which of the legal transformations to actually perform. It picks the final optimized program from the various programs represented in the saturated representation. Here, program expression graphs (PEGs) are employed, which is an intermediate representation designed specifically for equality reasoning. As far as compiler optimization is concerned, several operations are performed at this juncture. Common sub-expression elimination (CSE) reduces the number of duplicated computations by reusing previously defined and still available non-trivial expressions. If the same expression is computed in two different program points, CSE eliminates one of the computations, by replacing the second operation by an access to the register containing the result of the first evaluation. CSE is similar to constant propagation, in that the transformation is triggered by conditions represented by an equality between a register and an expression. In constant propagation this expression corresponds to a constant value, whereas in CSE it may be a more complex expression (involving arithmetic operators). The process of converting to and from PEGs produces optimizations well beyond constant propagation and CSE. It includes loop *invariant branch* hoisting and sinking, and several other operations.

[4] Lazy constraints are constraints not specified in the constraint matrix of the MIP problem, but must not be violated in a solution. It is used to speed up the solution algorithm.

[5] GUROBI is a set of high-end libraries for math programming, particularly for MIP and LP.

[6] MOSEK is a large-scale optimization software that solves linear, quadratic, general convex and mixed integer optimization problems.

[7] The NEOS server is the first network-enabled problem-solving environment for a wide class of applications in business, science, and engineering. The server is designed as a generic application service provider. Users submit a problem and their choice of an optimization solver over the Internet. The NEOS server computes all information (for example, derivatives and sparsity patterns) required by the solver, links the optimization problem with the solver, and returns a solution.

[8] For stochastic programming, a two-stage planning horizon is one where immediate (Here and Now) decisions ($\mathbf{x}_1$) have to be taken before all the problem elements have become known. Once this happens there are further, second-stage decisions ($\mathbf{x}_2$) to be taken according to the newly discovered events. So for the expectation we combine the probability-weighted minima of all the second-stage models, the resulting formulation of the problem is known as the Deterministic Equivalent (DEQ). It was observed in the resulting model form is precisely the form solvable by Benders' decomposition, the dual of Dantzig-Wolfe decomposition. In this method a solution $\mathbf{x}_1$ allows a subsequent dual-solutions to be calculated and applied to form an aggregated 'cut', which is a constraint added - thus giving a new solution $\mathbf{x}_1$, and so an iterative process is developed. Theory shows that the iterations converge to precisely the solution of the deterministic equivalent (DEQ) model.

[9] SMPS is a standard input format for multi-period stochastic programs based on MPS.

[10] Automake scans the package's “configure.in” to determine certain information about the package. Some autoconf macros are required and some variables must be defined in “configure.in.” Automake will also use information from “configure.in” to further tailor its output.

[11] POPT stands for Interior Point OPTimizer. Pronounced I-P-Opt, it is a software package for large-scale nonlinear optimization.

[12] Microsoft Solver Foundation is a new .NET-based optimization platform that includes a variety of solvers.

[13] In MS Windows applications programming, OLE Automation is an inter-process communication mechanism based on Component Object Model (COM) that was intended for use by scripting languages. It provides an infrastructure whereby applications called automation controllers can access and manipulate shared automation objects that are exported by other applications. In OLE Automation the automation controller is the “client” and the application exporting the automation objects is the “server.”

[14] RAD Game Tools is a privately-held company that develops video and computer game software technologies which are licensed primarily by video game companies.

[15] In object-oriented programming and software engineering, double dispatch is a mechanism that dispatches a function call to different concrete functions depending on the runtime types of the two objects involved in the call.

REFERENCES

Albright, S. C. (2010). *VBA for Modelers*, 3rd ed. Mason, Ohio: Thomson South-Western

Booth, D. T.; Cox, S. E.; Berryman, R. D. (2006). "Point Sampling digital imagery with 'Samplepoint'." *Environmental Monitoring and Assessment* 123, No. 1–3:97–108.

Bruce, B. (2007). A survey of open source geospatial software. Mapping and Geomatics Branch, Manitoba Conservation, Winnipeg, Manitoba, Canada.

Buckshaw, D. (2010). "Decision analysis software survey." *OR/MS Today* 37, No. 5, October.

Fourer, R. (2009). "Software survey: Linear programming." *OR/MS Today* 35, No. 3, June.

Fourer, R.; Ma, J.; Martin, K. (2010). "Optimization services: A framework for distributed optimization." *Operations Research* 58:1624–1636.

Galati, S. R. (2006). *Geographic information systems demystified*. Boston and London: Artech House.

Hillier, F. S.; Lieberman, G. J. (2009). "Chapter 21 – The art of modeling with spreadsheets." In *Introduction to operations research*, 9th ed. New York: McGraw-Hill.

Le Reverend, B. (2006). "Octave, a free MATLAB clone, and a bit more." retrieved from MacResearch Website in December 2009.

Maxwell, D. T. (2008). "Decision analysis: Find a tool that fits." Decision Analysis Software Survey Series, *OR/MS Today* 37, No. 5, October.

Nash, S. G. (1998). "Software survey: Nonlinear programming." *OR/MS Today* 25, No. 3, June.

Partyka, J.; Hall, R. (2010). "Vehicle routing software survey: On the road to connectivity." *OR/MS Today* 37, No. 1, February.

Pawlowicz, L. (2009). "Free Geography Tools" Website.

Prastacos, P. (1992). "Integrating GIS technology in urban transportation planning and modeling." *Transportation Research Record* 1305:125–130.

Sodhi, M. S. (2009). "Software review: Mathematica 7." *OR/MS Today* 36, No. 6, December.

Swain, J. J. (2009). "Statistical software survey: A long way from flip charts." *OR/MS Today* 36, No. 1, February.

Swain, J. J. (2011). "Software survey: Statistical analysis." *ORMS Today* 38, No. 1, February.

Tarrazo, M. (2006). "MATLAB: Software answers needs of users involved in numerical and symbolic analysis." Software Review series, *OR/MS Today* 35, No. 3, June.

Tate, R.; Stepp, M.; Tatlock, Z.; Lerner, S. (2009). "Equality Saturation: A New Approach to Optimization." *Proceedings of the 36th Annual ACM SIGPLAN – SIGACT Symposium on Principles of Programming Languages*, Savannah, Georgia.

Vanderzee, D.; Singh, A. (1995). "Survey of geographical information system and image processing software." *International Journal of Remote Sensing* 16, No. 2:383–389.

Winston, W. L.; Albright, S. C. (2007). *Practical management science,* 3rd ed. Mason, Ohio: Thomson/South-Western.

Synthesis Exercises and Problems

These exercises are carefully selected to complement the self-instructional modules, homework exercises, examples and case studies documented in the main body of the book. In many ways, they also supplement these varied illustrations of the concepts advance in this text; and the answers in the Solution Manual posted on my web site can be thought of as extensions to the main body of the book. Rather than a simple regurgitation of the basic computations, these synthesis exercises generally require a bit of thought, and many are open-ended case studies. To provide an integrated view, all the synthesis exercises were placed here in this section, rather than at the end of each chapter. The exercises and problems are categorized under the following topics:

I. Remote Sensing and Geographic Information Systems
II. Facility Location
III. Simultaneous Location and Routing
IV. Activity Derivation, Competition, and Allocation
V. Land Use Models
VI. Spatial-Temporal Information
VII. Term Project

We view this as a way to cut across all chapters in the book, emphasizing the main themes that run through this entire volume. For those who are more comfortable with examples (rather than concepts), the Solutions Manual on the web site will serve as a primer on the topics. (Contact the author at ychan@alum.MIT.edu for information about his web site. Students and professionals should enter in the SUBJECT block: "Request for sample solutions." Instructors should enter: "Request for Instructor's Guide.") The exercises also provide the opportunity to try out the software that comes with this book.

I. REMOTE SENSING AND GEOGRAPHIC INFORMATION SYSTEMS

This first group of problems range from the classic Bayesian classifier to image processing schemes such as histogram processing on the Training System/Image Processing (TS-IP) software, which is "Image" on the book's software CD. Two exercises on the Iterative Conditional Mode algorithm are included, illustrating a well-recognized classification technique. A problem is specifically introduced here to illustrate the prescriptive district clustering model advanced in this text. We then finish with a combined classification scheme in which the multicriteria

Y. Chan, *Location Theory and Decision Analysis*, 2nd ed., DOI 10.1007/978-3-642-15663-2,

decision-making procedure is explicitly incorporated as an integral part of the algorithm, showing that judgment is part and parcel of remote sensing and geographic information systems.

A. Bayesian Classifier

The Bayesian classifier is one of the ways to group pixels into different patterns—thus the classifier decides that pixel *j* belongs to a lake while pixel *i* belongs to a forest. We have illustrated in the "Bayesian Decision-Making" section of Chapter 3 how a decision boundary x_0 can be arrived at when there is only one attribute *x* such as a pixel's gray value. The concept can be extended to the case when there is more than one attribute for classification (say *n* attributes). The equations used in this context are as follows the (Gonzalez and Woods 1992). First the Gaussian distribution is extended to multidimensions by

$$P(\mathbf{x} \mid \mathbf{z}_j) = \frac{1}{(2\pi)^{n/2} \mid \mathbf{C}_j \mid^{1/2}} \ \exp\,[1-1/2(\mathbf{x} - \boldsymbol{\mu}_j)^T \mathbf{C}_j - 1(\mathbf{x} - \boldsymbol{\mu}_j)] \qquad \text{(E.1)}$$

where $\boldsymbol{\mu}$ is the mean vector and $\mathbf{C}$ is the covariance matrix respectively defined as

$$\boldsymbol{\mu}_j = \frac{1}{N_j} \sum_{\mathbf{x} \in G_j} \mathbf{x} \qquad C_j = \frac{1}{N_j} \sum_{\mathbf{x} \in G_j} \mathbf{x}\mathbf{x}^T - \boldsymbol{\mu}_j \boldsymbol{\mu}_j^T \qquad \text{(E.2)}$$

where N_j is the number of pattern vectors from class G_j (i.e., the number of pixel vectors belonging to class *j*), and the summation is taken over these vectors. The multidimensional decision boundary now looks like

$$d_j(\mathbf{x}) = \ln P(\mathbf{z}_j) - \frac{n}{2} \ln 2\pi - \frac{1}{2} \ln \mid \mathbf{C}_j \mid - \frac{1}{2} [(\mathbf{x} - \boldsymbol{\mu}_j)^T \mathbf{C}_j^{-1} (\mathbf{x} - \boldsymbol{\mu}_j)] \qquad \text{(E.3)}$$

Of course, the second term is equal for all cases, and may be subsequently dropped.

Now consider a two-dimensional reading for a 3 × 3 set of borings monitoring a groundwater pollution plume, with the gray values shown in italics (Wright and Chan 1994c),

	y-coordinate		
	1	2	3
x-coordinate			
1	*2*	*3*	*4*
2	*3*	*8*	*7*
3	*1*	*7*	*9*

Can you delineate the analytic and precise boundary of the plume based on the above set of equations?

B. Iterative Conditional Mode Algorithm

The iterative conditional mode (ICM) algorithm was described in detail in Chapter 6, under the "Contextual Allocation of Pixels" section. As demonstrated in the numerical example, a β of 0 produces a non-contextual classification, while increasing β accentuates the contextual bias. There is a tradeoff between β and σ^2, where σ^2 is the variance of pixels in a certain class. The σ should be small enough to prevent greatly overlapping regions, and at the same time β will need to be adjusted for the noise level of the image. A 6 × 6 grid of gray values is given below, with high values representing polluted groundwater and low values representing unpolluted water. Noise is introduced into the data by virtue of the data gathering procedure. For example, a value which has the same approximate gray value as the unpolluted groundwater exists in the center of pixels that are evidently polluted. The second 6 × 6 data set below shows a 3 × 3 area of apparently polluted ground water with possible noise on one of the sides of the 3 × 3 area. Also a single pixel (noise) with a pollution range gray value exists among unpolluted pixels.

3	5	4	3	4	2		3	5	4	3	4	2
3	4	3	2	3	3		3	4	8	6	7	3
4	2	4	8	4	3		4	2	7	8	5	3
5	3	10	9	8	12		5	3	10	9	8	4
3	4	9	4	7	7		3	6	4	4	5	5
2	5	11	12	10	9		2	4	5	4	4	4

Please perform the classification using the ICM on both of these two data sets (Wright and Chan 1994c).

C. Weighted Iterative Conditional Mode Algorithm

In this exercise (Wright and Chan 1994c), the weighted ICM algorithm (rather than the unweighted one used above) is to be applied to illustrate a couple of points. For the second data set above, the noise pixel in the polluted area could be classified as polluted water should a low enough β value be applied, since three of the five neighbors of the pixel are first-order neighbors. It can be shown also that the noise in the unpolluted area would be easier to discern using a weighted procedure. Notice the implementation is almost identical in both the weighted and unweighted cases. The only difference lies in the calculation of the "compare" value in which the summation must be broken into a first-order and a second-order summation. Now carry out the weighted ICM algorithm.

D. District Clustering Model

Shown in Chapter 6 under the district clustering model is a set of noninferior solutions for a small image entitled "Multiple subregion noninferior solutions." Examine the file labeled S2_4a_4b and S2_4a_5b in Figure 6.28, the first of these two code names stands for two subregions, the second and third suggest that an area of four pixels for subregions 1 and 2. The *a* and *b* entries specify two different variations on the boundary of the subregion, generating different non-inferior solutions. The two noninferior images are drawn below sequentially, where the bolded gray values stand for one subregion and the italicized stand for another:

11	**5**	**6**		*11*	*5*	*6*
8	**12**	**2**		*8*	**12**	**2**
5	*1*	*1*		*5*	**1**	**1**

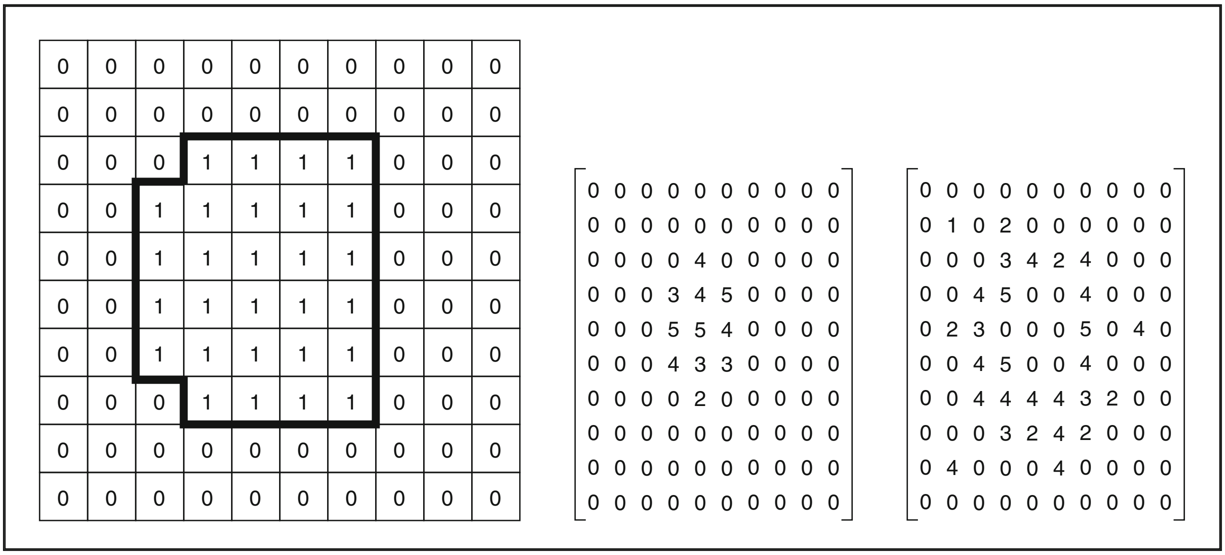
Using the "multiple subregion model" outlined in Chapter 6 in the subsection under the same name (Section X-B),

(a) Show the constraint-reduced feasible region model that generated these images;
(b) Verify step by step that we have generated the entire non-inferior solution set;
(c) Show the equivalent weighted objective function model.

E. Combined Classification Scheme

In monitoring groundwater pollution, measurements are made at wells placed discretely around the study area. Interpolation (such as kriging) has been made between these readings, forming a pixel map of the pollution level throughout the study area. Figure E.1 shows a well located at the center of the symmetrical cluster of readings. At the same time, remotely sensed data are available for the entire area. The ground truth data are given as well in Figure E.1.

Figure E.1 GROUND TRUTH, WELL DATA, AND REMOTELY SENSED DATA



0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	1	1	1	1	1	0	0	0
0	0	1	1	1	1	1	0	0	0
0	0	1	1	1	1	1	0	0	0
0	0	1	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$$\begin{bmatrix} 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&4&0&0&0&0&0\\ 0&0&0&3&4&5&0&0&0&0\\ 0&0&0&5&5&4&0&0&0&0\\ 0&0&0&4&3&3&0&0&0&0\\ 0&0&0&0&2&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0 \end{bmatrix} \qquad \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0\\ 0&1&0&2&0&0&0&0&0&0\\ 0&0&0&3&4&2&4&0&0&0\\ 0&0&4&5&0&0&4&0&0&0\\ 0&2&3&0&0&0&5&0&4&0\\ 0&0&4&5&0&0&4&0&0&0\\ 0&0&4&4&4&4&3&2&0&0\\ 0&0&0&3&2&4&2&0&0&0\\ 0&4&0&0&0&4&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0 \end{bmatrix}$$

Can you combine the two sources of information to delineate the pollution pattern more accurately than you would from a single source? Specifically, perform the following:

(a) Employ the ICM algorithm of Chapter 6 with due consideration to proximity as a factor. An inverse relationship is hypothesized between distance and importance in determining the allocation of some internal pixels (i.e., pixels not at the border or fringe of the image). For internal pixels, weights are scaled against eight neighbors. Assuming unitary distance separation between the subject pixel and its first-order neighbor, and a distance of $\sqrt{2} = 1.4142$ with its second-order neighbors. Thus the weight for first-order neighbors is 1.1716 and 0.8284 for second-order neighbors. The sum over all of its neighbors is $(4)(1.1716) + (4)(0.8284) = 8$ and the first-order neighbor is $1.1716/0.8284 = 1.4142$ times as important as the second-order neighbor in determining allocation of a pixel as specified initially.

(b) Employ multicriteria-optimization techniques as outlined in Chapter 5. Define in the decision space a binary variable that labels each pixel as being polluted when the variable is unitary valued. The two criteria in the outcome space—namely the value of the data and the value of contextuality in the ICM classification—are captured by the ground truth and the choice of the β value in the ICM algorithm respectively. Here, β is a measure of forced contiguity, applied parametrically for a 0–1 ranged weight for combining the two sources of information. The two data sets are shown in Figure E.1. Since the water is directly sampled there, one may wish fully to trust the data at the well, thus at the well the weight is unitary for the well reading and zero for the remotely sensed data. Also shown in the same figure is the ground truth, representing a subjective judgment by the decision maker.

(c) Determine the noninferior solutions that identify the most viable image classifications. A preference structure can be adopted whereby the smaller the deviation from the ground truth the more it represents a non-dominated solution. Zero deviation is considered Pareto optimal. Deviation in this case is defined as the number of pixels in the ICM-generated solution that are different from the ground truth. Likewise, the less the need for forced contiguity (i.e., the smaller the β value), the better.

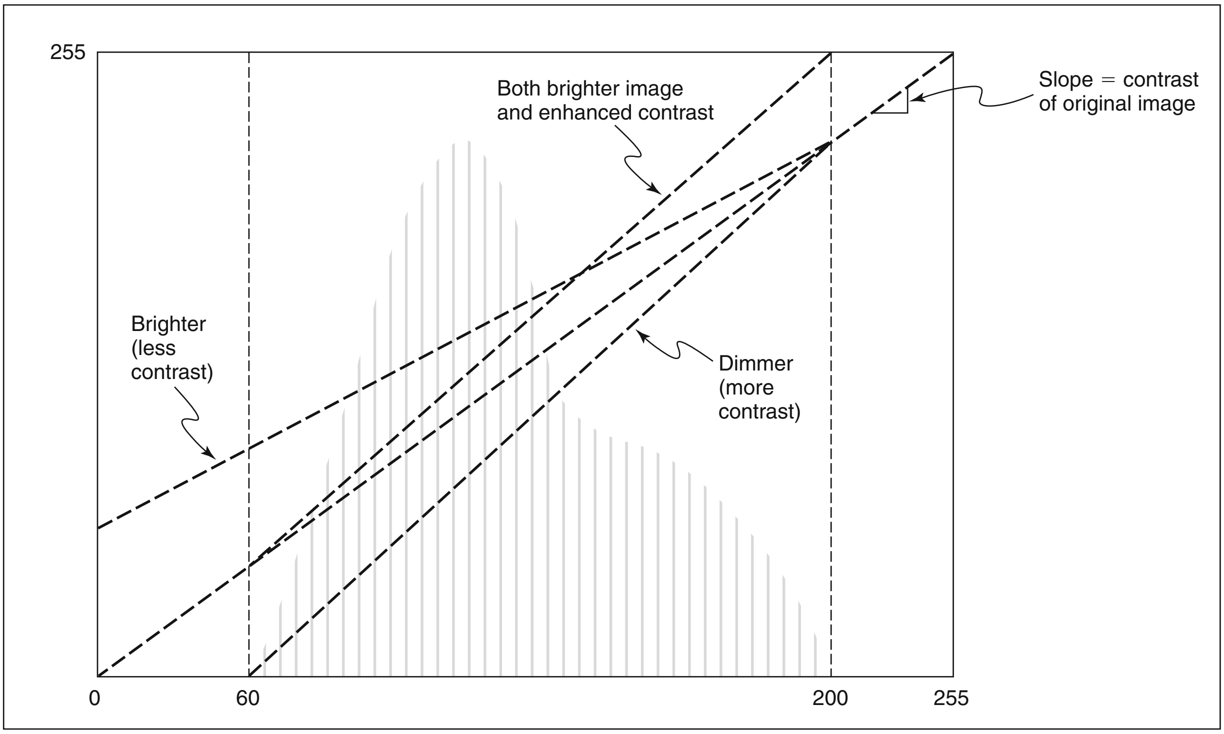
F. Histogram Processing

Refer to the TS-IP software under the SPACE directory, located on the CD/DVD that accompany this book. The image brightness histogram shows the number of pixels in the image having each of the 256 possible monochromatic values of stored brightness (Russ 1998; Gonzalez & Woods 1992). Peaks in the histogram correspond to the more common brightness values, which often identify particular structures that are present. Valleys between the peaks and the two tails indicate brightness values that are less common in the image. The flat regions at the two ends of the histogram show that no pixels have those values, indicating that the image brightness range does not necessarily cover the full 0–255 range available. Similarly, the pixels at the two tails of the gray values tend to contain noise, rather than the real image. Figure E.2 shows an example of such a histogram.

In order for the available gray levels to be used efficiently on the display, some will have to be removed (such as those at the two tails of the given histogram). It might be better to spread out the displayed gray levels in the peak areas selectively, compressing them in the valleys (or the two tails) so that the same number of pixels in the display shows each of the possible brightness levels. This is called **histogram equalization** or **histogram stretch**. Histogram equalization reassigns the brightness values of pixels. Individual pixels retain their brightness order (i.e., they remain brighter or darker than other pixels) but the values are shifted, so that an equal number of pixels have each possible brightness value. In many cases, this spreads out the values in regions where different regions meet, showing details in areas with a high brightness gradient. The equalization makes it possible to see minor variations within regions that appear nearly uniform in the original image. In this example, we show that the range 60–200 can be stretched out to occupy the entire spectrum, resulting in a dimmer image, but with better contrast.

The process is quiet simple mathematically. For each brightness level *j* in the original image (and its histogram), the newly-assigned relative-value *k* is calculated as $k = \Sigma_{i=0}^{j} N_i / n'$, where the summation counts the number of pixels in the image (by integrating the histogram) with brightness equal to or less than *j*. N_i is the number of pixels in the *i*th brightness level, and n' is the total number of pixels (or the total area of the histogram). This is graphically represented as the

Figure E.2 SEVERAL OPTIONS IN HISTOGRAM EQUALIZATION



dashed straight line plotted from the extreme left gray value to the extreme right value, representing the gray value range we wish to examine in detail.

In Figure E.2 are shown several ways to perform histogram equalization, including controlling brightness and contrast. Using the TS-IP software provided with the CD, please show on the Pentagon image these various options:

(a) dimmer with more contrast,
(b) brighter with less contrast, and
(c) both brighter and with contrast enhanced.

II. FACILITY LOCATION

Facility-location modeling is a key component of this book. Here we cover some less than obvious applications of these models. Following the airport location examples used extensively in the book, we further illustrate the nodal optimality conditions prevalent in not only min-sum location models, but also min-max models as well. The opposite of min-max problems is the max-min problem, commonly found in obnoxious facility location, which includes solid waste facilities. Another challenging facility location model is the quadratic assignment problem, in which interaction between facilities take place.

A. Nodal Optimality Conditions

Consider the cities of Cincinnati and Dayton, Ohio connected by Interstate Highway 75. Cincinnati has a metropolitan population of 2 million and Dayton, 1 million. A regional airport is proposed to serve both cities. It is to be located on I-75 such that the total person miles (PMT) to travel between the two cities is to be minimized. We have shown in Chapters 1 and 4 that the optimal location is Cincinnati. This is an example of nodal optimality conditions.

(a) Per discussions in Chapter 4: if the airport is to be located on I-75 so that the total person decibels of noise pollution is to be minimized, where should the airport be built?

(b) Suppose accessibility and noise exposure are of equal concerns, where should the airport be located? Accessibility is defined as the total PMT while noise exposure is the total person decibel.

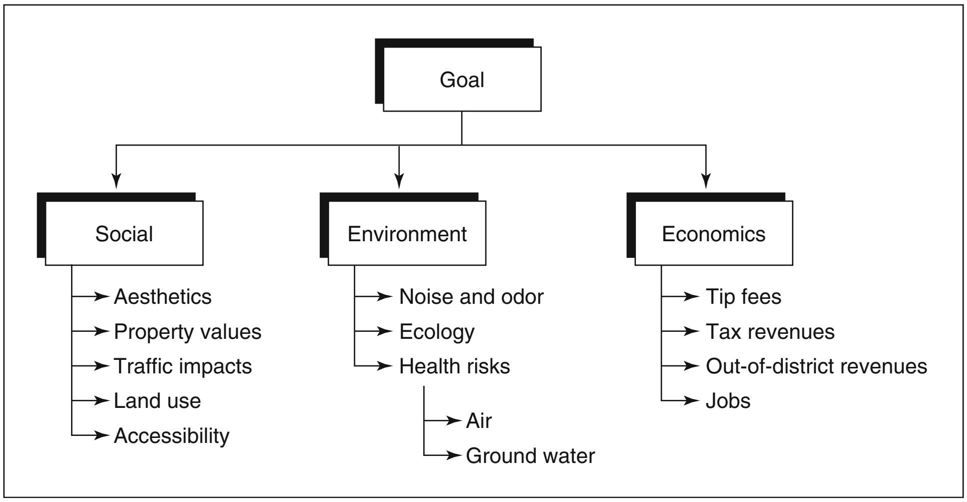
(c) Repeat questions (a) and (b) for the three-city case where Columbus is included. Columbus has a population of 2.1 million.

(d) Repeat the whole process for a four-city case in which Indianapolis is included in addition.

B. Solid Waste Facility

In locating a municipal solid waste facility, the analytic hierarchy process (AHP) has often been used. Junio (1994) proposed a hierarchy of attributes as shown in Figure E.3. Discuss the completeness and relevance of such a hierarchy definition. How would you quantify this hierarchy in executing AHP?

Figure E.3 HIERARCHY OF A MUNICIPAL SOLID WASTE PROBLEM



C. Quadratic Assignment Problem

Refer to the quadratic assignment problem as introduced in Chapter 4.

(a) Formulate the linearized version of model for the distance separation and flow interaction matrices as shown.
(b) Now solve this linear model.
(c) Is there anything peculiar about the solution to the linear model? If not, simply give the optimal assignment and the objective function. If yes, explain the peculiarity and again give the optimal assignment and the objective function value.

III. LOCATION-ROUTING

The integration of facility location and service delivery is a key feature of this book. We use a simple telecommunication network maintenance problem to lay out the integration. First, we define a region to be served by a maintenance facility using the districting technique. Then we place the facility using the service facility location model, followed by an evaluation of the entire maintenance procedure through a user performance model. To solve a real world problem, the three steps are executed repeatedly in a districting, location, and evaluation triplet. Having laid out this background, we break the problem into the service delivery step and then the combined location routing step. The basic building block of both steps is the quantification of spatial separation. This is illustrated in terms of **Minkowski's metric**, which is also known as l_p-metric—as defined in Chapter 5 under the "Deviational Measures" subsection.

A. Districting

The next three problems demonstrate a solution algorithm for improving maintenance depot location and service delivery operations (Patterson 1995). Here in the first problem, we define the districts each depot is supposed to serve. The model is based upon enumeration and was adapted for network topology by Ahituv and Berman (1988):

$$\begin{aligned} &\text{Min } \Sigma_j C_j x_j \\ &\text{s. t. } \Sigma_j x_j = p \\ &\Sigma_j a_{ij} x_j = 1 \qquad \forall i \end{aligned} \tag{E.4}$$

where $x_j = 1$ if subnetwork j is selected to form a district and zero otherwise; $a_{ij} = 1$ if node (zone) i is an element of subnetwork j and zero otherwise; p is the number of districts or subnetworks desired, and the equity measure

$$C_j = \frac{|\Sigma_i f_i - 1/p|}{\alpha/p}, \tag{E.5}$$

$0 < \alpha < 1$; and f_i is the fraction of demand at node i.

The algorithm consists of two different phases: Phase I determines all easible subnetworks (districts) within the larger network, and Phase II determines the final subnetworks based upon our equity objective-function in Equation E.5. Contiguity and compactness will be bounding constraints for the first phase. One final requirement is that the p subnetworks must be collectively exhaustive and mutually exclusive. In other words, every node must be within one and only one subnetwork. This is accounted for in Phase II.

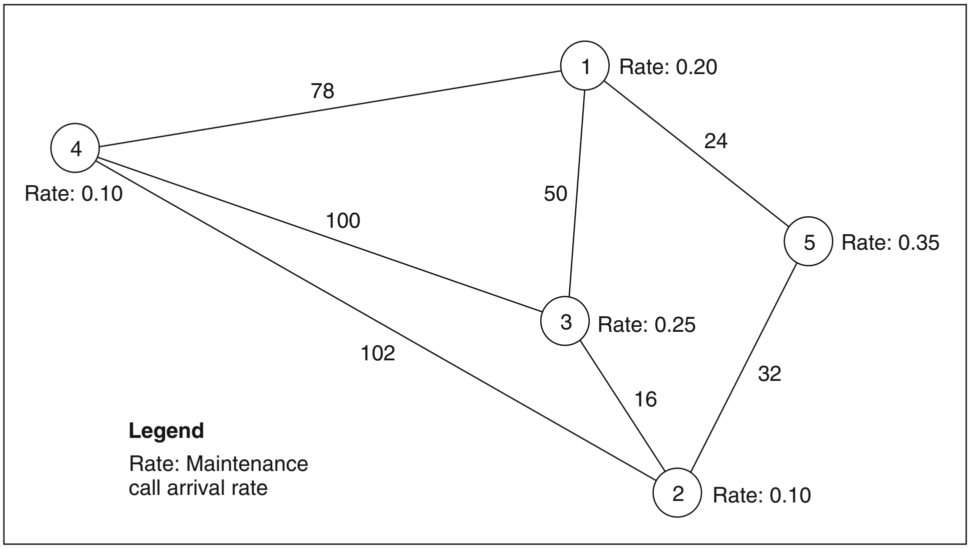
PHASE I: Using a tree search algorithm, we find the feasible set by picking the smallest number node and connecting contiguous nodes while enforcing the compactness requirement until the combined demand becomes redundant. Care must be taken to avoid creating separate enclaves, which are node(s) that are incapable of being separate subnetworks and cannot connect to other subnetworks without going through a previously defined subnetwork. This will prevent impossible solutions.

PHASE II: The algorithm for node partitioning was developed by Garfinkel and Nemhauser (1970). The following notation is needed: X is the set of fixed variables; $|X|$ is the number of fixed variables; D'' is the set of nodes in the districts, or zones of X; J is the set of districts in the current partial solution; N_j are the nodes J; and $|\cdot|$ is the cardinality of the set $\cdot$ in general.

The computational steps are briefly outlined below:

Step 1: Initialization. Set counter $|L| = 0$, and set $J = X$, $N_j = D''$.
Step 2: Choosing the next list. Pick the smallest number node not in N_j.

Figure E.4 SERVICE NETWORK



SOURCE: Patterson (1995). Reprinted with permission.

Step 3: Updating set J. Add the node to form subnetworks.

Step 4: Testing for a solution. Test $|L| = |J| - |X|$. If $|L| = 0$ stop, else $|L| = |L| - 1$.

Step 5: Finding a solution. Pick the largest cost subnetwork in J as the current solution. Go to Step 2.

Now for the network shown below in Figure E.4, please perform the districting procedure with $\alpha = 0.1$ to arrive at two service regions.

B. Minkowski's Metric

Consider two points $\mathbf{y}^1 = (14, 13)$ and $\mathbf{y}^2 = (4, 4)$ in a two-dimensional space. Employing the following general measure of deviation between $\mathbf{y}^1$ and $\mathbf{y}^2$, $r(\mathbf{y}; p) = [\Sigma_i |y_i^1 - y_i^2|^p]^{1/p}$, explore the behavior of numerical values of r for parameter p changing from 1 to ∞:

(a) Draw a diagram of function $r = f(p)$. What are the general properties of such a function?

(b) Perform the same analysis for p changing from 0 to 1 and also from $-\infty$ to 0. Do these cases show any meaningful interpretation?

(c) Perform a more difficult but very rewarding exercise: Do these distance measures, especially for p between 1 and ∞, correspond to any particular subfamily of utility (or value) functions? Can you identify such a subclass?

(d) Perform the following graphic exercise: Define a point $\mathbf{y}^2 = (0, 0)$ in a two-dimensional space. Plot all such points $\mathbf{y}^1$ whose distance from $\mathbf{y}^2$ is equal to a fixed number r^*, that is, $r = r^*$. Choose $r^* = 1$ and draw such loci of points $\mathbf{y}^1$ for p ranging from 1 to ∞. (Pay special attention to $p = 1, 2, \infty$). Do the resulting "shapes" suggest any connection with utility functions?

(e) Are there some points in (*d*) which have the same distance from point $\mathbf{y}^2$ regardless of the value of *p*? What are the other characteristics and possible interpretations of such points?

IV. ACTIVITY DERIVATION, ALLOCATION AND COMPETITION

The transition from facility location to land use models can be marked by activity derivation, allocation, and competition. Thus economic activities such as population and employment are generated at an activity center. Residential neighborhoods then compete to provide housing for these people, resulting in a distribution of residents among these neighborhoods. Here in this group of exercises, we solve a matrix multicriteria game, in which there is more than one payoff among the competitors. The gravity model is a traditional way to analyze competition among geographic areas. Using the gravity versus transportation models exercise, one can see that the gravity model is an extension of the "all or nothing" assignment of activities. Assignment from one single supply exclusively to one single demand is performed by the Hitchcock-Koopman transportation model. This is complemented by the calibration of a doubly constrained model.

A. Multicriteria Game

Consider the following game decision maker 1 (DM1) maximizes his minimum gain while decision maker 2 (DM2) minimizes her maximum loss. Gain of DM1 is exactly equal to the loss of DM2 (i.e., a zero-sum game). Instead of the single metric used in the conventional payoff matrix, there is more than one criterion in measuring payoffs. These multiple payoffs are therefore expressed in terms of a vector (rather than a scalar). An example appears below in Table E.1, where the cells contain the two payoffs for each pair of decisions reached between DM1 and DM2:

Thus if both DMs decide to play their second option, DM1 wins 3 units in the first criterion and 2 in the second. DM2 loses the same number. The symbols p'_i and q'_j denote the probability DM1 and DM2 will play the *i*th and *j*th strategy

Table E.1 MULTICRITERIA GAME

		DM2		
		q'_1	q'_2	q'_3
	p'_1	(3, 2)	(3, 4)	(1, 5)
DM1	p'_2	(2, 1)	(3, 2)	(2, 2)
	p'_3	(4, 1)	(1, 3)	(3, 1)

respectively. When p' and q' assume fractional values, the game is a called a **mixed strategy** game. A **pure strategy** is when p's and q's are 1 or 0 in value.

Let each vector payoff $\boldsymbol{a}_{ij} = (\boldsymbol{a}^1_{ij}, a^2_{ij})^T$ be replaced by a convex combination of both components: $wa^1_{ij} + (1 - w), a^2_{ij}$, where w is a 0–1 ranged weight. For example, $a_{11} = w3 + (1 - w)\,2 = w + 2$, and so on. It can be shown that, similar to a conventional zero-sum two-person game an LP can be set up to solve this problem, where the primal and dual solutions correspond to the strategy taken by the two decision makers. If nonnegative variables p and q are defined such that $p' = pz'$ and $q' = qz'$, the equivalent LP is:

$$\begin{array}{ll}
\text{Max} & q_1 + q_2 + q_3 \\
\text{s.t.} & \\
& (w+2)\,q_1 + (4-w)\,q_2 + (5-4w)\,q_3 \le 1 \\
& (w+1)\,q_1 + (w+2)\,q_2 + 2\;q_3 \le 1 \\
& (3w+1)\,q_1 + (3-2w)\,q_2 + (2w+1)\,q_3 \le 1
\end{array}$$

(a) Solve this LP by varying the weights w from 0 to 1.

(b) Is there an equilibrium—defined here as a pair of decisions with which both sides are happy? At this equilibrium, z' is a nonegative number representing the gain to DM1 and the loss to DM2.

B. Gravity versus Transportation Model

Refer to the doubly constrained gravity model as discussed in the subsection bearing the same title in Chapter 3. When the value of α becomes 1 in the propensity function $F(C_{ij})$, the function becomes a special function of travel cost, $F(C_{ij}) = C_{ij}^{-\alpha} = C_{ij}^{-1}$, and the doubly constrained gravity model can be written as

$$\begin{aligned}
V_{ij} = (k_i l_j V_i V_j)\,F(C_{ij}) = z'_{ij}\,C_{ij}^{-1} \;\text{ or }\; z'_{ij} = C_{ij}\,V_{ij} \\
z = \Sigma_{ij}\,z'_{ij} = \Sigma_{ij}\,C_{ij}\,V_{ij} \\
\sum_{i=1}^{n'} V_{ij} = V_j \quad j = 1, 2, \ldots, n' \\
\sum_{j=1}^{n'} V_{ij} = V_i \quad i = 1, 2, \ldots, n'
\end{aligned} \tag{E.6}$$

z in Equation (E.8) is interpreted as the total travel cost now. By minimizing total travel cost (for instance, veh-min), we have the classical transportation model. Notice this model reflects the system optimum rather than user optimum as obtained by conventional gravity-model calibration. Now answer the following questions:

(a) What value would α assume in the propensity function to have maximum accessibility?

(b) What value would α assume to have minimum accessibility?

(c) For a prescriptive model, what is the resulting trip distribution for case (a)?

(d) For a prescriptive model, what is the trip distribution for case (b)?
(e) Interpret the result of (c) and (d).

C. Calibration of a Doubly Constrained Model

Given the following data on interzonal trips V_{ij} and the associated costs C_{ij}, please calibrate a doubly-constrained gravity model:

$$[V_{ij}] = \begin{array}{rcc} & 1 & 2 \\ \text{zone 1} & 100 & 200 \\ \text{zone 2} & 300 & 50 \end{array} \text{ and } [C_{ij}] = \begin{array}{rcc} & 1 & 2 \\ \text{zone 1} & 1 & 2 \\ \text{zone 2} & 2 & 1 \end{array}$$

Suppose $F(C_{ij}) = c_{ij}^{-2}$, carry out the calculations as far as you can, following the procedure described in the doubly constrained model subsection of Chapter 3. Give the final four equations for the four unknowns, and solve the equations.

V. LAND USE MODELS

Analysis of land use models is a center piece of this book. Here, the economic-base and activity distribution exercise shows how the activity derivation, distribution, and competition concepts can be used to simulate the housing requirements of a college town over time. This set of calculations is then formalized in the iterative Lowry model calculation, which is encoded on the software CD.

A. Economic-Base and Activity Allocation

In a study of a college town, State College, Pennsylvania, Chan and Rasmussen (1979) forecasted housing requirements. Using the basic concepts of the Lowry model, they derived the subareal housing requirement of the town using the university enrollment as the basic activity. Their algorithm follows a two-part procedure:

Part I. Housing Demand Factor

1. Define the zoning types of all residentially zoned developable land.
2. Establish the number of students, blue-collar employees, and white-collar employees from tract i working at employment center c—labeled here as E_{ic}^{S}, E_{ic}^{B}, E_{ic}^{W} respectively.
3. Determine the separation d between each tract centroid and employment center.
4. Obtain the percentage of student, blue-collar, and white-collar commuters traveling a distance of d miles to the related employment center—labeled $f_i^{S}(d)$, and $f_i^{B}(d)$, and $f_i^{W}(d)$ respectively.
5. Determine the percentage of students, blue-collar workers, and white-collar workers in residential type t—labeled p_t^{S}, p_t^{B} and p_t^{W}, respectively
6. Compute the housing demand factor: $V_{it}^{d} = \Sigma_k \Sigma_c \Sigma_d E_{ic}^{k} f_i^{k}(d)\, p_t^{k}$.

Part II. Allocation of Housing Demand

1. Determine the excess housing supply in tact i, ΔN_i. The excess is equally distributed among the number of zoning types t_{Max}: $\Delta N_{it} = \Delta N_i / t_{\text{Max}}$.
2. Determine the maximum holding capacity for developable dwelling units: N_{it}^c = (developable average) (average dwelling units per acre).
3. Allocate the total housing demand N to each tract i: $N_{it} = NV_{it}^d / \Sigma_i \, \Sigma_t V_{it}^d$. The housing demand for housing type t in tract i can either be accommodated by the excess housing supply ΔN_{it} or new construction. Housing demand exceeding the holding capacity of a tract would have to be located elsewhere. The additional developable capacity of a tract for housing type t is $\Delta N_{it}^c = N_{it}^c - N_{it} - \Delta N_{it}$.
4. Additional iterations are necessary as long as one or more ΔN_{it}^c is negative (i.e., there is spill over from a tract), and excess capacity still exists in the region to accommodate the excess. Otherwise, the algorithm terminates.

Chan and Rasmussen then compared their forecast with the ones by the Centre Region Planning Commission (CRPC). The housing projection performed by the CRPC is computed by a two-step procedure: (1) The future population for the region is computed; and (2) the number of dwelling units is derived from that figure. The derivation process is generally founded on a extrapolation forecasting techniques. The CRPC population forecast takes into consideration a cohort survival model and a straight-line proportional model. (These techniques are discussed in the "Econometrics Modeling" section of Chapter 2.) The following assumptions are made among both studies: (a) No substantial in or out migration would take place, which implies the student enrollment at Penn State University would stabilize at 31,500 by 1985. (b) Existing trends, including birthrates/death rates and other coefficients and ratios, will remain constant over time for each township of the Centre Region.

Since the study, the dwelling units that were actually observed became available. These figures are tabulated beside the Chan and Rasmussen and CRPC forecasts in Table E.2. Can you perform a "before-and-after" analysis as to the accuracy of the forecasts by the Chan-Rasmussen model vis-a-vis the CRPC study?

Table E.2 COMPARISON OF FORECASTS AND OBSERVED HOUSING UNITS

1985 Figures		College	Ferguson	Halfmoon	Harris	Patton	State College
Single family	Chan & Rasmussen	1599	2105	221	960	1434	3316
	CRPC	1599	2505	276	1145	1801	3114
	Observed	1785	2209	303	1124	1768	2650
Multiple family	Chan & Rasmussen	208	544	6	21	790	6207
	CRPC	351	591	6	31	849	6545
	Observed	545	964	19	176	1730	7837

B. Forecasting Airbase Housing Requirements

Now that you are familiar with the Chan-Rasmussen housing model, can you use the same model to forecast housing requirements for an Air Force base? Similar to the college town model, this new model is based on the hypothesis that the foundation of the local economy is an Air Force base (Bahm et al. 1989). Whiteman Air Force Base (AFB)—near Knob Noster, Missouri—is chosen for the study. Whiteman was picked because the base is a major source of employment for the region and was expected to grow at the time of the study in 1989. The source of the increase in military- and civilian-employment is the new B-2 bomber wing.

Three types of economic activities are envisioned to increase: military and their dependents, civilian Department of Defense (DOD) employees, and civilian non-DOD employees. There are 25 housing tracts or zones in the region. There are four employment centers: Warrensburg, Sedalia, Knob Noster, and Whiteman AFB. Commuting distance is measured in one-mile (1.6-km) increments, with the longest commuting distance being 46 miles (73.6 km). There are five residential types: single family, double family, multiple family, dormitories and non-residential. Additional developable capacities, excess housing, and resident profiles are documented in Table E.3. The information is listed by each tract/zone *i*. By resident profile we mean the percentage of military, DOD civilian, and non-DOD civilians in each type of housing—whether it be single family, double family, multiple family, or dormitory. Commuting distances from each of the 25 tracts/zones to the four employment centers are shown in Table E.4. The trip distribution, or the percentage of workers traveling distance *d* to an employment center, is shown in Table E.5. Included in the table are the increases in military, DOD civilian and non-DOD civilian jobs in each of the four employment centers.

Now forecast the housing requirements at the study area based on these assumptions: (a) insignificant projected increase in employment from manufacturing in Warrensburg and Sedalia, (b) insignificant projected increases in employment or student enrollment at Missouri State University, and (c) only a small amount of associated cross-commuting from Whiteman to other points in the study region. All these make Whiteman AFB the major employer in the projected future, attracting the local population to the base.

VI. SPATIAL-TEMPORAL INFORMATION

The unifying theme throughout this book is really how one analyzes spatial-temporal information in general. In this last block of problems, we let the data guide us in the analysis. The first problem eloquently shows the difference between spatial and univariate forecasts, particularly regarding their respective accuracies. Subsequently we worry about the calibration of a spatial forecasting model, an area so demanding that much more research is still needed.

Table E.3 ADDITIONAL DEVELOPABLE CAPACITIES, EXCESS HOUSING* AND RESIDENT PROFILE

Tract/zone	Single family	Double family	Multiple family	Dormitory
1	0 (2)	0 (2)	–	892
2	181	210	562	–
3	0	0	0	0
4	88	93	92	–
5	45	–	–	–
6	39	–	–	–
7	2490	2495	2492	–
8	24	–	–	–
9	20	27	932	–
10	35	–	–	–
11	38	–	–	–
12	36	–	–	–
13	30 (9)	–	–	–
14	30	–	–	–
15	30 (9)	–	–	–
16	20	–	–	–
17	141	135	893	–
18	35	–	–	–
19	2496	2500 (6)	2500 (2)	–
20	25	–	–	–
21	30 (9)	–	–	–
22	0	0	0	0
23	50 (7)	–	–	–
24	50 (4)	–	–	–
25	31	–	–	–

Resident profile	Single family	Double family	Multiple family	Dormitory
% Military	0.290	0.320	0.073	0.317
% Civilian/DOD	0.645	0.040	0.315	0
% Civilian/non-DOD	0.616	0.031	0.353	0

*Excess housing numbers are in parentheses.

Table E.4 COMMUTING DISTANCES[a] TO EMPLOYMENT CENTERS

Tract/zone	Warrensburg	Sedalia	Knob Noster	Whiteman AFB
1	12	20	3	1
2	10	18	1	3
3	8	22	3	3
4	17	10	8	10
5	25	10	15	17
6	13	23	7	9
7	7	24	4	6
8	17	23	12	10
9	1	28	10	12
10	9	30	17	17
11	7	27	17	19
12	8	32	15	17
13	16	43	23	25
14	20	45	30	30
15	14	43	26	26
16	20	46	30	30
17	28	1	18	20
18	22	12	15	15
19	23	7	14	16
20	24	17	15	17
21	38	15	25	26
22	35	10	23	24
23	35	8	25	26
24	35	8	25	26
25	37	12	28	28

[a] In integral miles (or multiples of 1.6 km).

A. Cohort Survival Method

The cohort survival method is an econometric technique introduced in Chapter 2, in the "Interregional Growth and Distribution" subsection. Please review the discussions in the text and answer these questions (Jha 1972):

(a) Suppose these statistics are gathered for York County, Pennsylvania during the 1940–1945 period. The number of births is 2,000 and the number of deaths is 500. The average population for the period is 210,000. There were 1,400 people migrating to York and 1,295 migrating out. Define the following terms for a certain forecast time period: crude birthrate, crude death rate, and net migration.

Table E.5 TRIP DISTRIBUTION AND JOB PROFILES AT THE EMPLOYMENT CENTERS

Employment distribution	Military	Civilian/DOD	Civilian/non-DOD
Warrensburg	400	0	400
Sedalia	500	0	800
Knob Noster	100	0	100
Whiteman AFB	2045	355	241
Trip distribution	**Military**	**Civilian/DOD**	**Civilian/non-DOD**
1	0.2	0	0.55
2	0.05	0	0.1
3	0.2	0.08	0.05
4	0.01	0.05	0.03
5	0.01	0.01	0.02
6	0.01	0.05	0.02
7	0.01	0.01	0.02
8	0.01	0.01	0.02
9	0.01	0.01	0.02
10	0.01	0.01	0.02
11	0.08	0.01	0.02
12	0.165	0.25	0.01
13	0.01	0.01	0.01
14	0.01	0.01	0.01
15	0.01	0.01	0.01
16	0.0015	0.01	0.01
17	0.025	0.01	0.01
18	0.05	0.01	0.01
19	0.25	0.01	0.01
20	0.14	0.37	0.01
21	0.005	0.02	0.01
22	0.001	0.01	0.01
23	0.001	0.01	0.01
24	0.001	0.005	0.001
25	0.001	0.005	0.001
26	0.0005	0.005	0.001
27	0.0005	0.005	0.001
28	0	0.001	0.001
29	0	0.001	0.001
30	0.0005	0.001	0.001
31	0	0.001	0.0005
32	0	0.001	0.0005
33	0	0.001	0.0005
34	0	0.001	0.0005
35	0	0.001	0.0005
36	0	0.005	0.0005
37	0	0.005	0
38	0	0.005	0
39	0	0.005	0

(b) Check the population for females in 1945 in York County, Pennsylvania by the cohort survival method. Use the population statistics shown in the following table. Note that these numbers are in hundreds, i.e., 10 means 1000.

Age	0–4	5–9	10–14	15–19	20–24	25–29	Total
1940	10	14	15	18	22	24	103
1945	7	10	12	14	16	21	80

The entries in this table represent the number of people in each age group. The surviving ratio of the 0–4 year group is 98% and the percentage of female children is 49%. The fertility rate of the 15–19 year group is 43%; the rate for 20–24 groups and 25–29 groups is 56%.

VII. TERM PROJECT

As a capping stone, the purpose of this term project is to

(a) Show how Multi-criteria Decision Analysis can be used in spatial information technology such as image processing.

(b) Demonstrate the practical use of a Bayesian classification model coded in MATLAB and provided in the book CD/DVD under the PATTERN directory.

(c) Demonstrate via real-life example that is downloaded from the NOAA GOES satellite dish using the GVAR image-acquisition software. Alternatively, a real-life image can be obtained from the default sample image called TESTG, or other sources, including the collection of satellite images on the book CD under the folder IMAGEFILES.

Step 1.

Organize the class into individual teams as follows:

Team 1	Name 1 Name 2 Name 3
Team 2	Name 4 Name 5 Name 6
Team 3	Name 7 Name 8 Name 9
Team 4	Name 10 Name 11 Name 12
Team 5	Name 13 Name 14 Name 15
Etc.	

Professional responsibility dictates that each team member participates.

Step 2.

Please review the following documents:

1. Read Chapter 3, Section VII.E, in our textbook on "Bayesian classifier," and try to understand these concepts.

   (a) Use an observed distribution to estimate the underlying distribution of a set of data.
   (b) Review a bimodal distribution of gray-value intensities as a precursor to the Forest vs. Lake classification example.
   (c) Relate the methodology to the Forest-Lake example in the following reading assignment.

2. Read Chapter 6, Section VIII, in our textbook on "Pattern Recognition," and try to understand these concepts:

   (d) Spectral Classification vs. Spatial/Contextual classification
   (e) Spectral Classification example
   (f) Spatial Classification example

3. Read Chapter 6, Section IX, in our textbook on "District Clustering Model," and try to understand these concepts:

   (g) The Benabdallah-and-Wright model
   (h) Apply the model to analyze the Washington DC Mall image.

Step 3.

1. Now answer the trailing questions briefly and to the point. The only exception is Question 4, in which *annotated* output of the computer runs are required, both in hard copy and software copy.
2. Write a technical paper to systematically document the theories, methods, as well as the results based on a satellite imagery provided as part of the software. In other words, embed answers to the questions in Part 1 as a complete technical essay. The paper should be typewritten and submitted in both hard and soft copies.

Part I of the Project

REFER TO FIGURE 6.24, ENTITLED "CONTEXTUAL VS. NON-CONTEXTUAL IMAGE CLASSIFICATION."
Question 1: The above Figure illustrates the difference between *spectral vs. contextual classification* of an image. Can you explain these two terms in your own language?

REFER TO FIGURE 6.23, ENTITLED "PDF IN BAYESIAN CLASSIFIER."
Question 2: We used a Bayesian classifier to perform a combined spectral and contextual classification. Explain in your own terms how this is applied toward the example on Lake vs. Forest pixels as shown by the previous and following illustrations.

Figure E.4 $k = 2$ IMAGE CLUSTERING

Raw image

0	0	0	0	0	1
0	2	2	2	1	0
1	1	4	4	2	0
0	2	5	3	2	0
0	2	1	2	1	1
0	0	0	1	0	0

Image processed by *k*-medoid algorithm for two clusters

2	2	2	2	2	2
2	2	2	2	2	2
2	2	1	1	2	2
2	2	1	1	2	2
2	2	2	2	2	2
2	2	2	2	2	2

REFER TO THE TWO-CLASS "42-PIXEL IMAGE" EXAMPLE WORKED OUT IN CHAPTER 6 SECTION VIII.B, ENTITLED "CONTEXTUAL ALLOCATION OF PIXELS," and also FIGURE 6.33, ENTITLED "SPOT SUB-IMAGE GRAY VALUES"

Question 3. Aside from Bayesian decision theory, multi-criteria optimization has been employed to perform image classification, as illustrated in Figure 6.33 on SPOT image of the Washington DC Mall. Can you explain the results as shown in the three frames for Channels 1, 2, and 3?

Part II of the Project

REFER TO FIGURE E.4, ENTITLED "K = 2 IMAGE CLUSTERING."

Question 4. Under the directory PATTERN, a MATLAB program has been provided on the CD/DVD to perform the *k*-medoid classification based on multi-criteria optimization and Bayesian decision theory, as illustrated in the above Figure E.4 example. Based on this demonstration of a *k*-Medoid Algorithm, please run the *k*-Medoid software, IMGKMED, for the GOES satellite image "TESTG" as provided on the CD/DVD. Assisted by the trailing instructions, please

(a) Execute the algorithm for $k = 3$ until it converges. Explain how you know it converges. In the MATLAB IMGKMED algorithm, the placement of your "seed" medoids is random. Correspondingly, you would expect the number of iterations to reach convergence different from one run to another, even for the same initial image.

(b) Perform three runs corresponding to the following weight sets and discuss the differences between your results.

(c) Discuss the possible application of the *k*-medoid classification technique in terms of preventing natural hazards such as storms.

In addition to the annotated outputs, please make sure you answer each of the five questions in your technical paper.

Weight set	w_1	w_2
1	0.2	0.8
2	0.5	0.5
3	0.8	0.2

Operating Instructions for the MATLAB program "IMGKMED"

- □ Open the MATLAB software.
- □ Go to the directory/folder in which you have placed the IMGKMED folder.
- □ At the command line, enter IMGKMED**.**
- □ Press the "Load image" button.
- □ Select the "TESTG.BMP" file.
- □ Set $k = 3$.
- □ Set the weights for "proximity" or for "Channel 1" (monochromatic grayscale) by moving the lever of the weight scale.
- □ Determine the number of iterations required to reach convergence.
- □ Execute by pressing the "Cluster" button.

Notice that since the provided image(s) is black and white, channels 2 and 3 are nonfunctional. It is suggested that you leave the setting for "Proximity" at the half way point when you start out. For simplicity, please leave the channels 1, 2 and 3 settings to the middle point (50–50), and only change w_1 and w_2 (the "proximity" setting). When you run the IMGKMED software, make sure you re-load the original "test" image every time. In other words, anytime you change your input parameters, such as the number of iterations, you need to re-load the "test" image. Otherwise, erratic behavior will result from the IMGKMED software.

Question 5. Pick a downloaded, image-processed satellite image other than the "testg.bmp" file. Can you speculate how that image is actually constructed? Please simply pick the IVHRR image in the IMAGEFILES folder on the CD/DVD and analyze it. (By the way, the disk that comes with the book has a lot more images. The disk also contains an image-processing software called TS-IP, complete with a User's Manual.)

REFERENCES

Ahutuv, N.; Berman, O. (1988). *Operations management of distributed service networks—A practical quantitative approach.* New York: Plenum Press.

Bahm, P.; Ross, M.; Chan, Y. (1989). Forecasting housing requirements for an Air Force installation. Working Paper. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson Air AFB, Ohio.

Banaszak, D.; Cordeiro, J.; Chan, Y. (1997). Using the K-medoid and covering approaches in pattern-recognition problems. Working Paper. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson Air AFB, Ohio.

Burnes, M. D. (1990). Application of vehicle routing heuristics to an aeromedical airlift problem. Master's Thesis. (AFIT/GST/ENS/90M-3). Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson Air AFB, Ohio.

Chan, Y. (2005) *Location transport and land-use: Modeling spatial-temperal information*. Berlin and New York: Springer.

Chan, Y.; Rasmussen, W. (1979). "Forecasting housing requirements in a college town." *Journal of the Urban Planning and Development Division* (American Society of Civil Engineers) 105:9–23.

Clough, J.; Millhouse, P.; Chan, Y. (1997). The obnoxious facility location problem. Working Paper. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson Air AFB, Ohio.

Daskin, M. (1995). *Network and discrete location: Models, algorithms, and applications*. New York: Wiley.

Francis, R.; McGinnis, L.; White, J. (1999). *Facility layout and location: An analytical approach*, 3rd ed. Englewood Cliffs, New Jersey: Prentice-Hall.

Garfinkel, R. S.; Nemhauser, G. L. (1970). "Optimal political redistricting by implicit enumeration techniques." *Management Science*, 16:495–508.

Gonzalez, R. C.; Woods, R. E. (1992). *Digital image processing*. Reading, Mass.: Addison-Wesley.

Grosskopf, S.; Magaritis, D.; Valdmanis, V. (1995). "Estimating output substitutability of hospital services: a distance function approach." *European Journal of Operational Research* 80:575–587.

Irish, T.; May, T.; Chan, Y. (1995). A stochastic facility relocation problem. Working Paper. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson Air AFB, Ohio.

Jha, K. (1972). Demographic models. Working Paper. Department of Civil Engineering. Pennsylvania State University. University Park, Pennsylvania.

Junio, D. F. (1994). Development of an analytic hierarchy process for siting of municipal solid waste facilities. Master's Thesis. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson Air AFB, Ohio.

Kanafani, A. (1983). *Transportation demand analysis*. New York: McGraw-Hill.

Mandl, C. (1979). *Applied network optimization*. New York: Academic Press.

Memis, T.; Eravsar, M.; Chan, Y. (1997). Integer programming solution to Route Improvement Synthesis and Evaluation (RISE). Working Paper. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson Air AFB, Ohio.

Mitchell, E. J. (1969). "Some econometrics of the Huk rebellion." *American Political Science Review* 63:1159–1171.

Patterson, T. S. (1995). Dynamic maintenance scheduling for a stochastic telecommunications network: Determination of performance factors. Master's Thesis. Department of Operational Sciences. Graduate School of Engineering. Air Force Institute of Technology. Wright-Patterson Air AFB, Ohio.

Pfeifer, P. E.; Bodily, S. E. (1990). "A test of space-time ARMA modelling and forecasting of hotel data." *Journal of Forecasting* 9:255–272.

Piskator, G. M.; Chan, Y. (1997). Estimating a production function and efficient frontier for the United States Army recruiting battalions. Working Paper. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson Air AFB, Ohio.

Russ, J. C. (1998). *Image processing handbook*, 3rd ed. Boca Raton: CRC Press.

Steppe, J. M. (1991). Locating direction finders in a generalized search and rescue network. Master's Thesis. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson Air AFB, Ohio.

Steuer, R. E. (1986). *Multiple criteria optimization: Theory, computation, and application*. New York: Wiley.

Tapiero, C. S. (1971). "Transportation-location-allocation problems over time" *Journal of Regional Science* 11:377–384.

Wright, S. A. (1995). Spatial time-series: Pollution pattern recognition under irregular intervention. Master's Thesis. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson Air AFB, Ohio.

Wright, S. A.; Chan, Y. (1994a). A network with side constraints for the k-medoid method for optimal plant location applied to image classification. Working Paper. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson Air AFB, Ohio.

Wright, S. A.; Chan, Y. (1994b). Multicriteria decision-making applied to the ICM contextual image classification technique. Working Paper. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson Air AFB, Ohio.

Wright, S. A.; Chan, Y. (1994c). Pure and polluted groundwater classification on a pixel map. Working Paper. Department of Operational Sciences. Air Force Institute of Technology. Wright-Patterson Air AFB, Ohio.

Zelany, M. (1982). *Multiple criteria decision making*. New York: McGraw-Hill.

Appendix 1

Control, Dynamics, and System Stability

While the main body of the text concentrates on Location Theory and Decision Analysis, there are some computational aspects of model solution that the readers may wish to review. Four appendices are provide here for that purpose. The first appendix follows our self-instructional module on "Empirical Modeling". In this appendix, we review the basic theories that govern the evolution of complex systems, wherein systems transition from one state to another over time. Systems may evolve on their own or external influence may be brought to bear upon their development. In both cases, there can be smooth transitions as well as precipitous happenings. We discuss the conditions under which a system may change between these two types of evolution—namely from smooth to precipitous changes and vice versa. Most importantly, we wish to effect these changes where we can, so as to direct the development toward a desired goal.

Stochastic, nonlinear system is a powerful tool for location theory and decision analysis. We have seen an example in Chapter 4 under the topic of "Optimal Control of Spatial Interaction." Other examples can be found under "Economic Base Theory," "Facility Expansion," and "Competitive Location and Games." These are scattered throughout this book and the accompanying CD/DVD. For the curious, Chan (2005) applies the methodology in depth while discussing the "Garin-Lowry Model" and "Spatial Equilibrium".

I. CONTROL THEORY

The concept of control theory was introduced in Appendix 3, where an example of inventory control was worked out in the context of vehicle dispatching. In the example, trucks deliver a stock of cargo $X(t)$ at the loading dock over the afternoon between hours t_0 and t_1. The cargo is to be airlifted to a destination. We wish to construct a schedule to minimize operating cost and schedule delay. The problem was solved by discrete dynamic programming, wherein the optimal dispatch schedule as indicated by the control variable $U(t)$ is determined. Here we will generalize and ormalize the results in a more systematic way using control theory (Silberberg 1990).

The general form of a control theory problem is expressed as a maximization problem instead of minimization:

$$\max_{U(t)} \int_{t_0}^{t_1} f(X(t), U(t), t)\, dt \qquad \text{(A1.1)}$$

subject to the state equation[1]

$$\dot{X}(t) = g(X(t), U(t), t) \tag{A1.2}$$

with end-point conditions $X(t_0) = X_0$, $X(t_1) = X_1$ (or $X(t_1)$ free) and some control set $\{U(t)\}$ or the set of decision variables. The time between t_0 and t_1 is called the planning period. In many important problems, t_1 tends to infinity, or the planning horizon is far out into the future. End-point conditions vary depending on the problem context. Typically the initial stock of the state variable X is fixed, although the final stock may not be. In addition, there may be restrictions on the variables, such as non-negativity on the state X, and perhaps inequality bounds on the control variable U. An example of such inequality bounds is the 0–1 valued dispatch or hold policy $0 \leq U \leq 1$ at each time the dispatch decision is renewed.

Let us ignore for the moment how the control problem is solved, but assume that finite interior solution $(U^*(t), X^*(t))$ does exist, in other words, a time path that leads from $X_0(t_0)$ to $X_1\ (t_1)$ as defined by the control set $\{U(t)\}$. The value $(U^*(t), X^*(t))$ represents the optimal time paths of the control variable U and the state variable X. Although we are suppressing it in the notation, X^* and U^* in fact depend on the initial parameters X_0, t_0, and so forth. Thus in the cargo dispatching example, the amount of cargo at the dock at the starting time t_0 defines the ultimate dispatching policy. Denote the resulting value of the objective functional (objective function) as $F(X_0, t_0)$, that is

$$F(X_0, t_0) = \int_{t_0}^{t_1} f(X^*(t), U^*(t), t)\, dt \tag{A1.3}$$

where a functional $f(\cdot)$ is defined as a function that has a domain whose elements are functions, sets, or the like, and that assumes numerical values. Although Equation A1.1 requires us to find an actual path as specified by the function $(U^*(t), X^*(t))$, this maximizes an integral function, which once found results in some ordinary maximum expressed in terms of the parameters of the model. (Notice we suppress t_1 here in Equation A1.3, as the parameter is not germane to the present discussion).

Given the initial state X_0, a marginal value of the stock exists for any time t between the initial time t_0 and the terminal time t_1. Denote this imputed value by the equivalent of the Lagrange multiplier (or the dual variable) $\lambda(t) = F_X(X^*(t), t)$, where F_X stands for the derivative of F with respect to X. This marginal value of the stock, $\lambda(t)$, is often referred to as the costate or adjoint variable in control theory. The change in the value of the stock caused by dispatching is $d[\lambda(t)X(t)]/dt = \lambda\dot{X} + X\dot{\lambda}$. The true net benefit of dispatching at some schedule $U(t)$ is the sum of the benefits in the present, $f(X, U, t)$, and the change in the maximum value of the stock caused by executing that schedule in the present. The optimal path is obtained by always setting the true marginal net benefits equal to zero along the entire optimal path of values $(U^*(t), X^*(t))$. Thus, we can characterize this solution to the control problem as requiring that, at each time instance t ($t_0 \leq t \leq t_1$), the first derivative of F with respect to t be maximized, or for a continuous function F

$$\max_{U, X} [\, f(X, U, t) + \lambda\dot{X} + X\dot{\lambda}\,] \tag{A1.4}$$

Using the state Equation A1.2, this becomes

$$\max_{U, X} [\, f(X, U, t) + \lambda g\ (X, U, t) + X\dot{\lambda}\,] \tag{A1.5}$$

We suppress the dependence on t at this point because we have not yet found the functions ($U^*(t)$, $X^*(t)$) and expressed them as functions of t. Differentiating with respect to the control variable U and the state variable X yields

$$f_U + \lambda g_U = 0 \tag{A1.6}$$

$$f_X + \lambda g_X + \dot{\lambda} = 0 \tag{A1.7}$$

Equation A1.6 is called the maximum principle; Equation A1.7 is called the **costate** or **adjoint equation**. These two conditions plus the state equation $\dot{X} = g(X, U, t)$ are the necessary conditions for an optimal path ($U^*(t)$, $X^*(t)$) of control and state variables over the planning period. Also determined is the path of marginal values of the stock, $\lambda(t)$.

Equations A1.6 and A1.7 are generally expressed in terms of the expression $H = f + g$, called the **Hamiltonian.** The maximum principle is $\partial H/\partial U = 0$ (assuming an interior solution to the problem) while the adjoint equation is $\partial H/\partial X = -\dot{\lambda}$. In the original problem, given the initial stock level, X_0, choosing $U(t)$ determines the state equation $\dot{X}(t)$ and thus the state variable $X(t)$. There is really only one independent variable, U, in this control problem. However, the introduction of the new variable $\lambda(t)$ adds another degree of freedom; as in static Lagrangian analysis, we pretend the problem has one more dimension than it actually has.

Using the maximum principle, Equation A1.6, which is not a differential equation (in other words, equation containing derivatives of t), and invoking the implicit function theorem of calculus, we can solve for U: $U = k\ (X, \lambda, t)$. Substituting this into the state and adjoint equations produces two first-order equations and

$$\dot{X} = g(X, k(X, \lambda, \mathrm{t}), t) \tag{A1.8}$$

and

$$\dot{\lambda} = -f_X\,(X, k(X, \lambda, t), t) - \lambda g_X\,(X, k(X, \lambda, t), t) \tag{A1.9}$$

Solving these differential equations (and using the relevant end-point conditions to evaluate the constants of integration) yields the optimum path of X and λ. Using the solutions to these equations—by substituting them into $k\,(X, \lambda, t)$—yields the optimum path of the control variable, U. The reader can see the close parallel between control theory and dynamic programming as explained in Appendix 3.

Example
Consider the optimal control problem

$$\text{Max } \int_0^t (-X - \frac{1}{2}\,\alpha U^2)\, dt$$

subject to $\dot{X} = U$, $X(0) = X_0$, $X(1) = X_1$, where $\alpha > 0$ is a parameter for this problem. The Hamiltonian for this problem is $H(X, U, \lambda) = -X - \frac{1}{2}\alpha U^2 + \lambda U$. Assuming an interior solution, the necessary conditions are $\partial H/\partial U = -\alpha U + \lambda = 0$ and

$\partial^2 H/\partial U^2 = -\alpha = 0$. By assumption $\alpha > 0$, so $\partial^2\mathrm{H}/\partial U^2 < 0$. Solving $\partial H/\partial U = 0$ for U gives $U = \lambda/\alpha$. The other necessary conditions are the state and adjoint equations $\dot{X} = \partial H/\partial\lambda = U, \dot{\lambda} = -\partial H/\partial X = 1$. Using $U = \lambda/\alpha$ in these equations yields, $\dot{X} = \lambda/\alpha$, $X(0) = X_0$, $X(1) = X_1$, and $\dot{\lambda} = 1$. Integrating $\dot{\lambda} = 1$ directly gives $\lambda^*(t) = t + c_1$, where c_1 is an unknown (as of yet) constant of integration. Substitute $\lambda^*(t)$ in λ/α to get $\dot{X} = (t + c_1)/\alpha$. Integrating this equation yields $X^*(t) = t^2/2\alpha + c_1 t/\alpha + c^2$ where c_2 is another constant of integration. The constants of integration c_1 and c_2 are determined by using the initial and terminal conditions $X(0) = X_0$ and $X(1) = X_1$, respectively. Use $X(0) = X_0$ in $X^*(t)$ to get $X^*(0) = c_2 = X_0$. Now use $X(1) = X_1$ to obtain the value of c_1: $X^*(1) = \frac{1}{2\alpha} + c_1/\alpha + X_0 = X_1$; thus $c_1 = \alpha(X - X_0) - \frac{1}{2}$. These constants of integration are then substituted in (X^*, λ^*) to yield their optimal paths, and then λ^* is substituted into $U = \lambda/\alpha$ to give the control's optimal time path. Doing this gives the solution of

$$X^*(t;\alpha, X_0, X_1) = \frac{t^2}{2\alpha} + \left[(X_1 - X_0) - \frac{1}{2\alpha}\right] t + X_0 \tag{A1.10}$$

$$\lambda^*(t;\alpha, X_0, X_1) = t + \alpha(X_1 - X_0) - \frac{1}{2} \tag{A1.11}$$

$$U^*(t;\alpha, X_0, X_1) = \frac{t}{\alpha} + (X_1 - X_0) - \frac{1}{2\alpha} \tag{A1.12}$$

Notice how the state, control, and marginal values are all expressed as functions of one single variable, time t, in the final solution. ■

II. CALCULUS OF VARIATIONS

Let us consider a special case of the control problem where $\dot{X} = g(X, U, t) = U$. That is, the time rate of change of the stock is identical to the control variable, rather than some general function $g(\cdot)$ that might also include the stock itself and time. Simply put, control activity is in direct proportion to (and equal to) the rate of accumulation or depletion. Control in this case is the degenerate case of "going with the flow." Now substitute this state equation $U = \dot{X}$ into the integrand $f(\cdot)$ in Equation A1.1. The result is the following objective functional

$$\mathrm{Max} \int_{t_0}^{t_1} f(X, \dot{X}, t)\, dt \tag{A1.13}$$

We call this problem the calculus of variations (Silberberg 1990). In this problem, we determine a function $f(\cdot)$ such that a certain definite integral involving that function and certain of its derivatives takes on a maximum or minimum value. Notice this special case of the general control theory problem has been illustrated by the numerical example worked out above, where U is exactly set to $\dot{X}$. The corresponding solution maps out an optimal path as specified by the state variable $X(t)$ in Equation A1.10.

In this special case, the necessary conditions for a maximum (or minimum) are as follows. Remember the maximum principle is $H_U = H_{\dot{X}} = f_{\dot{X}} + \lambda g_{\dot{X}} = 0$. However, $g_{\dot{X}} = g_U \cong 1$ here, so this condition becomes

$$f_{\dot{X}\dot{X}} = -\lambda \qquad \text{(A1.14)}$$

Similarly the adjoint or costate equation is

$$H_X = f_X + \lambda g_X = f_X = -\dot{\lambda} \qquad \text{(A1.15)}$$

recognizing $g_X = \partial U/\partial X = \partial\dot{X}/\partial X \cong 0$. Since the right-hand side of the adjoint Equation A1.15 directly above is the time derivative of the right-hand side of Equation A1.14, these two equations can be combined into $df_X/dt = \partial f/\partial X$ Carrying out the differentiation in the above equation results in the equivalent expression

$$f_{\dot{X}\dot{X}}(X, \dot{X}, t) \equiv f_{\dot{X}\dot{}\,t} + f_{\dot{X}\dot{}\,X} + f_{\dot{X}\dot{}\,\dot{X}}\ddot{X} \qquad \text{(A1.16)}$$

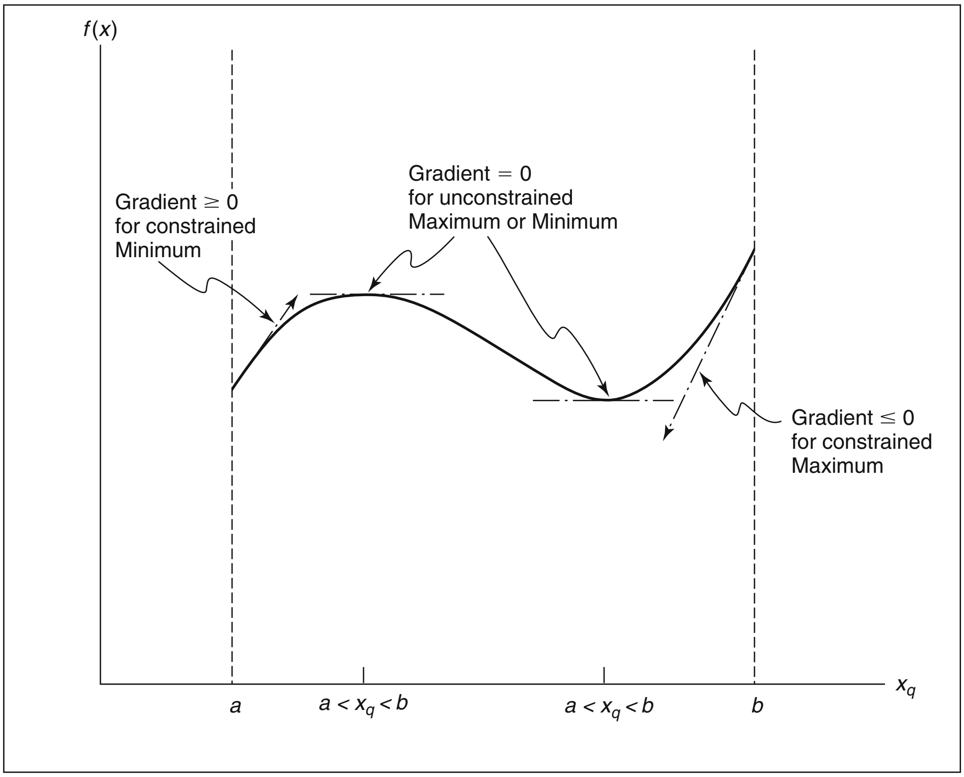
This is the classic Euler-Lagrange equation defining the necessary condition for an optimal path. Solutions of this equation are known as extremals and an extremal which satisfies the appropriate end conditions at t_0 and t_1 is called a stationary function. Application of Equation A1.16 results in a second-order differential equation (except for special cases), whereas the necessary conditions of control theory result in the first-order simultaneous Equations A1.8 and A1.9. There is no uniform computational advantage to one approach over the other. However, the Euler-Lagrange equation is difficult to interpret, while the control theoretic equations often provide useful characterizations of the dynamics of spatial economic models.

Example
Take the control theoretic numerical example shown in the above section, where $f(\cdot) = -X - \frac{1}{2}\alpha\dot{X}^2$. According to the Euler-Lagrange equation one can verify by regular calculus that $f_X = -\alpha\dot{X}$ where $f_X = -1$. Solving the second-order differential equation $\ddot{X} = 1/\alpha$ with the end-point conditions $X(0) = X_0$ and $X(1) = X_1$ yields the same solution as worked out previously. The solution is identical to Equation A1.10, as one would expect. ■

III. VARIATIONAL INEQUALITY

Obviously, both control theory and the calculus of variation are tools to analyze the optimality conditions of functionals. A general condition that encompass both of these techniques can be stated: Let $f(\mathbf{x})$ be a functional on a normed (regular) vector space Ω, it has a directional derivative at $\mathbf{x}_q$, and $\Omega_q \subset \Omega$ be convex. A necessary condition for $\mathbf{x}_q \in \Omega$ to be a maximum of f on Ω_q is that for all $\mathbf{x}_q \in \Omega_q$, the gradient $\nabla f^T(\mathbf{x}) = (\partial f/\partial x_1, \ldots, \partial f/\partial x_n)$ (or the generalization of the first derivative) of $f(\mathbf{x}_q, \mathbf{x} - \mathbf{x}_q)$ is less than or equal to a zero vector ($\leq \mathbf{0}$). This condition is illustrated for the maximization and minimization over a two-dimensional case in Figure A1.1. In the unconstrained case, the necessary optimality conditions are that the gradient of $f(\mathbf{x}_q, \mathbf{y})$ is equal to a zero vector ($\mathbf{0}$) for all $\mathbf{y}$ in Ω, where $\mathbf{y} = (\mathbf{x} - \mathbf{x}_q)$. We call these **variational equalities.** In the constrained case, the optimality condition is called **variational inequalities** (Minoux 1986). Though these results are straightforward to establish, they constitute the foundation of the calculus of variation. In fact it can be shown that they make it possible for us to derive the necessary optimality conditions known as the Euler-Lagrange equation for an interior (unconstrained) optimum.

Figure A1.1 ILLUSTRATION OF VARIATIONAL INEQUALITY



A. Fundamentals

We can provide a formalization of the above discussion: Let f be a smooth real valued function on the closed interval $\Omega_q = [a, b]$. We seek the points $x_q \in \Omega_q$ for which $f(x_q) = \text{Min}_{x\in\Omega_q} f(x)$ (Kinderlehrer and Stampacchia 1980). Three cases can occur for the two-dimensional case as shown in Figure A1.1: (a) if $a < x_q < b$, then $\dot{f}(x_q) = 0$; (b) if $x_q = a$, then $\dot{f}(x_q) \geq 0$, and (c) if $x_q = b$, then $\dot{f}(x_q) \leq 0$. These statements can be summarized by writing

$$\dot{f}(x_q)(\, x - x_q) \geq 0 \qquad \forall x \in \Omega_q \tag{A1.17}$$

Such a set of relationships will be referred to as variational inequality illustrated here in two dimensions.

Let f be a smooth real valued function defined on the closed convex-set Ω_q of Euclidean n-dimensional space. Again we shall characterize the points $\mathbf{x}_q \in \Omega_q$ such that $f(\mathbf{x}_q) = \text{Min}_{\mathbf{x}\in\Omega_q} f(\mathbf{x})$. Assume $\mathbf{x}_q$ is a point where the minimum is achieved and let $\mathbf{x} \in \Omega_q$. Since Ω_q is convex, the segment $(1 - w)\mathbf{x}_q + w(\mathbf{x} - \mathbf{x}_q)$, $0 \leq w \leq 1$, lies in Ω_q.The function $F(w) = f(\mathbf{x}_q + w(\mathbf{x} - \mathbf{x}_q))$, $0 \leq w \leq 1$, attains its minimum at $w = 0$. Analogous to the two-dimensional case above, $\dot{F}(0) = \nabla f^T(\mathbf{x}_q)(\mathbf{x} - \mathbf{x}_q) \geq \mathbf{0}$ for any $\mathbf{x} \in \Omega_q$. Consequently, the point $\mathbf{x}_q$ satisfies the variational inequality

$$\mathbf{x}_q \in \Omega_q\text{: } \nabla f^T(\mathbf{x}_q)(\mathbf{x} - \mathbf{x}_q) \geq 0 \qquad \forall \mathbf{x} \in \Omega_q \tag{A1.18}$$

If Ω_q is bounded, the existence of at least one $\mathbf{x}_q$ is immediate.

It should be noted that the above two cases—two and n-dimensional—can be solved by means of calculus since they depend on a finite number of variables. Many optimization problems have unknowns beyond a finite number of n variables. Consider a function $u(t)$ of the real variable t on some interval $[a, b]$. Since the graph of the function u is defined by infinite pairs of $[t, u(t)]$, we shall say that we are dealing with an optimization problem in infinite dimension. We have already encountered this in control theory, where $u(t) = U(t)$ is the control variable over time t. In this case, there are an infinite number of control paths $U(t)$ between the initialpoint $t = a$ and end-point $t = b$. More generally, we shall see that such problems can be formulated in the following way. Given a vector space Ω of (infinite dimension) and a functional f on Ω, find u^* such that for a minimization problem, $f(u^*) \leq f(u)$, $u \in \Omega$, for unconstrained optimization, or such that $f(u^*) \leq f(u)$, $u \in \Omega_q \subseteq \Omega$, for constrained optimization over a convex region. At this point, we will illustrate an application of variational inequality in an infinite dimensional space. The following example is similar to a problem of the calculus of variations.

Example

Let Ω be a bounded domain with boundary $\delta\Omega$ and let ψ be a given function on $\Omega = \Omega \cup \delta\Omega$ satisfying $\max_{\Omega}\psi \geq 0$ and $\psi \leq 0$ on $\delta\Omega$. Define $\Omega_q = \{y \geq \psi$ in Ω and $y = 0$ on $\delta\Omega\}$, where y is a function continuously differentiable in Ω. Notice this is a convex set of functions that we assume is not empty. We seek a function $u \in \Omega_q$ for which $\int_{\Omega} |\nabla u|^2 d\mathbf{x} = \underset{x \in \Omega_q}{\min} \int_{\Omega} |\nabla y|^2 d\mathbf{x}$.[2] Assuming such a y function to exist, we argue analogously to our previous discussion relying again on the convexity of Ω_q. For any $y \in \Omega_q$, the sequence $u + w(y - u) \in \Omega_q$, $0 \leq w \leq 1$, whence the function $f(t) = \int_{\Omega} |\nabla(u + w(y - u))|^2\, d\mathbf{x}$, $0 \leq w \leq 1$, attains its minimum at $w = 0$. This implies that $\dot{f}(0) \geq 0$, which leads to the variational inequality

$$u \in \Omega_q : \int_{\Omega} \nabla u^T\, \nabla(y - u)\, d\mathbf{x} \geq 0 \qquad \forall y \in \Omega_q \tag{A1.19}$$

Intervening here is the point set $\{\mathbf{x} \in \Omega$: $u(\mathbf{x}) = \psi(\mathbf{x})\}$. Its presence distinguishes u from the solution of a boundary value problem such as the end-point conditions imposed on the Euler-Lagrange second-order differential equation. As mentioned, one can interpret u as the height function of the equilibrium position of a thin membrane constrained to lie above the body $\{(\mathbf{x}, x_{n+1})$: $x_{n+1} \leq \psi(\mathbf{x})$, $\mathbf{x} \in \Omega\}$ and with fixed height zero ($\psi = 0$) on the boundary $\delta\Omega$. In spatial economics, we may have a market defined within a geographic boundary. The consumers in this market are charged a price of ψ, which is to be maximized. Conventional business practice dictates, however, that the price be as uniform as possible within the defined market such that market equilibrium results. ■

B. Existence and Uniqueness

Variational inequalities are general formulations that encompass a plethora of mathematical problems, including, but not limited to, optimization and complementarity problems. Variational inequalities were originally developed as a tool for the study of certain classes of partial differential equations (equations containing partial derivatives) such as those that arise in mechanics. A membrane example has been shown above. Such problems were defined over infinite dimensional spaces.

We focus here, however, on the finite-dimensional variational-inequality problem, mainly defined for economic equilibrium applications (Nagurney 1993).

In geometric terms, variational inequality, Equation A1.18, states that the gradient $\nabla f^{T}(\mathbf{x})$ is orthogonal to the feasible convex set Ω_q at the point $\mathbf{x}_q$. This formulation is particularly convenient because it allows for a unified treatment of equilibrium problems and optimization problems. For example, the variational inequality problem can be shown to contain the complementarity problem as a special case. The nonlinear complementarity problem, introduced earlier as part of the Karash-Kuhn-Tucker condition in Chapter 4, is a system of equations and inequalities stated as: Find $\mathbf{x}_q \geq \mathbf{0}$ such that

$$\nabla f(\mathbf{x}_q) \geq \mathbf{0} \ \text{ and } \ \nabla f^{T}(\mathbf{x}_q)\, \mathbf{x}_q = \mathbf{0} \tag{A1.20}$$

Whenever $\nabla f(\mathbf{x}) = \mathbf{A}'\mathbf{x} + \mathbf{b}$, where $\mathbf{A}'$ is an $n \times n$ matrix and $\mathbf{b}$ is an $n \times 1$ vector, Equation A1.20 is then known as the linear complementarity problem.

Variational inequality theory is also a powerful tool in the qualitative analysis of equilibria. Existence of a solution to a variational inequality problem follows from continuity of the function $\nabla f(\mathbf{x})$ entering the variational inequality, provided that the feasible set Ω_q is defined in the real space. It can be shown that variational inequality (Equation A1.18) admits a solution if and only if there exists a bounded solution $\mathbf{x}_q$. Qualitative properties of existence and uniqueness become easily obtainable under certain monotonicity conditions. For example, if $\nabla f(\mathbf{x})$ is strictly monotone on Ω_q, then the solution is unique, if one exists.

Monotonicity is closely related to **positive definiteness**, in other words, the generalization of a positive second derivative, where positive definiteness is defined to be the value of

$$\mathbf{x}^{T}\nabla^{2} f(\mathbf{x})\, \mathbf{x} = \mathbf{x}^{T}\left[\frac{\partial^{2} f(\mathbf{x})}{\partial x_i \partial x_j}\right]\mathbf{x} \tag{A1.21}$$

Let $\mathbf{x} = (\leftarrow x_i \rightarrow)^{T}$ be a vector of decision variables and $\mathbf{F}'(\mathbf{x}) = (\leftarrow F_i(\mathbf{x}) \rightarrow)^{T}$ be a vector of functions for $i = 1, \ldots, n$. These functions are characterized by asymmetric interactions $\partial F'_i(\mathbf{x})/\partial x_j \neq \partial F'_j(\mathbf{x})/\partial x_i\ (i \neq j)$. Suppose that $\nabla \mathbf{F}'(\mathbf{x}) = \dot{\mathbf{F}}'(\mathbf{x})$ is continuously differentiable on Ω_q. Let us further suppose the Jacobian matrix (or the generalization of the gradient for asymmetric interactions)

$$\dot{\mathbf{F}}'(\mathbf{x}) = \begin{bmatrix} \frac{\partial F'_1}{\partial x_1} \cdots \frac{\partial F'_1}{\partial x_n} \\ \cdot \qquad \cdot \\ \cdot \qquad \cdot \\ \cdot \qquad \cdot \\ \frac{\partial F'_n}{\partial x_1} \cdots \frac{\partial F'_n}{\partial x_n} \end{bmatrix}$$

which need not be symmetric, is positive semi-definite (or the expression A1.21 for $F'_j(\mathbf{x})$ is greater than or equal to zero). Then $\dot{\mathbf{F}}\boldsymbol{(x)}$ is monotone. If the function is positive definite (or expression A1.21 is strictly greater than zero), then $\dot{F}(\mathbf{x})$ is strictly monotone.

Example
Given the Jacobian matrix

$$\nabla \mathbf{F}'(\mathbf{x}) = \dot{\mathbf{F}}'(\mathbf{x}) = \begin{bmatrix} \dfrac{\partial F'_1}{\partial x_1} & \dfrac{\partial F'_1}{\partial x_2} \\ \dfrac{\partial F'_2}{\partial x_1} & \dfrac{\partial F'_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2x_1{}^2 + x_1 + 3x_2^2 + 6x_2 + 6 & \cdot \\ \cdot & \cdot \end{bmatrix}$$

and the Hessian is

$$\nabla^2 F'_1(\mathbf{x}) = \begin{bmatrix} \dfrac{\partial^2 F'_1}{\partial\, x_1{}^2} & \dfrac{\partial^2 F'_1}{\partial x_1 \partial x_2} \\ \dfrac{\partial^2 F'_1}{\partial x_2 \partial x_1} & \dfrac{\partial^2 F'_1}{\partial\, x_2{}^2} \end{bmatrix} = \begin{bmatrix} 4 & 0 \\ 0 & 6 \end{bmatrix}$$

$$\mathbf{x}^T \nabla^2 F'_1 \mathbf{x} = (x_1\ x_2) \begin{bmatrix} 4 & 0 \\ 0 & 6 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 4x_1{}^2 + 6x_2{}^2 > 0$$

excluding the origin (0, 0). If it can be shown that the Hessian of the other three entries of matrix $\dot{\mathbf{F}}'$ is also positive definite, then $\dot{\mathbf{F}}'$ is strictly monotone. If this is true for the entire convex feasible region Ω_q, then the solution to the corresponding optimization problem is unique. ■

IV. CATASTROPHE THEORY

Let **X** be a set of state variables describing some system—the dependent variables to be predicted in a model—and let **U** be a set of variables one can control. Then in a gradient system,[3] the equilibrium position is determined by

$$\min_{\mathbf{X},\,\mathbf{U}} f(\mathbf{X}, \mathbf{U}) \tag{A1.22}$$

This concept has been introduced in the previous section on control theory. The dynamics of the process is given by $\partial f / \partial \mathbf{X} = \nabla f$ and the minimum of f, of course, occurs when

$$\nabla f = \mathbf{0} \tag{A1.23}$$

The appearance of the gradient ∇, of the potential function f explains the name of this type of system (Wilson 1981; Lorenz 1993).

The solution to Equation A1.23 gives the equilibrium point, which minimizes the potential function in Equation A1.22. As **U** varies, this determines a surface in the space (**X**, **U**). This is a surface representing possible equilibrium states of the system. If, for example, there is a single state variable X and two control variables U_1 and U_2, then this will be a surface in the three-dimension space (X, U_1, U_2).

In this context, distinction is often made between a slow variable and a fast variable. Correspondingly, one may attach the time argument t behind the variables, which now read as $\mathbf{X}(t)$ and $\mathbf{U}(t)$. By convention $\mathbf{X}(t)$ is a fast variable and $\mathbf{U}(t)$ a slow variable. Correspondingly, one can think of **U** as a set of parameters influencing **X**. For a smooth, slow and small change in one or more of the **U** variables, a corresponding smooth change in the state variables **X** can be anticipated. For this to occur, the surface in (**X**, **U**) space of equilibrium solutions has to be itself smooth and not folded in any way. It has long been recognized that when, for a given **U**, there are multiple solutions for **X**, then something more complicated can occur. The essence of catastrophe theory is the classification of these complications and the proofs that, in a number of cases, they fall into a small group of basic types.

A. Basic Concepts

The solutions of Equation A1.22 or equivalently Equation A1.23 are the stationary points of the function f or, more precisely, of a family of functions of **X** (the fast variables), parameterized by **U** (the slow variables). Stationary points are often maxima or minima, which are distinguished by, in the single state variable case, the second derivative of f being negative or positive respectively. (In the multi-state variable case, the corresponding result is that the Hessian matrix or the generalized version of second derivative is negative or positive definite, respectively, as mentioned earlier in this appendix.) When stationary points are not maxima or minima, the second derivative is zero or the Hessian matrix is singular. Such equilibrium points are known as singularities and it is at and near such points that unusual system behavior is observed. What catastrophe theory does is to classify the kinds of singularities that can occur. It has been shown that, for a number of control variables in the vector **U** up to or equal to four, the types of singularities, in a topological sense, are relatively few. For example, in the case of a single state variable and two control variables, the surface of equilibrium points around a singularity must be topologically equivalent to the cusp surface, which is illustrated in Figure A1.2. Application of the cusp catastrophe is found in the "Chaos, Catastrophe, Bifurcation and Disaggregation" chapter of Chan (2005) under the "Spatial Dynamics" section.

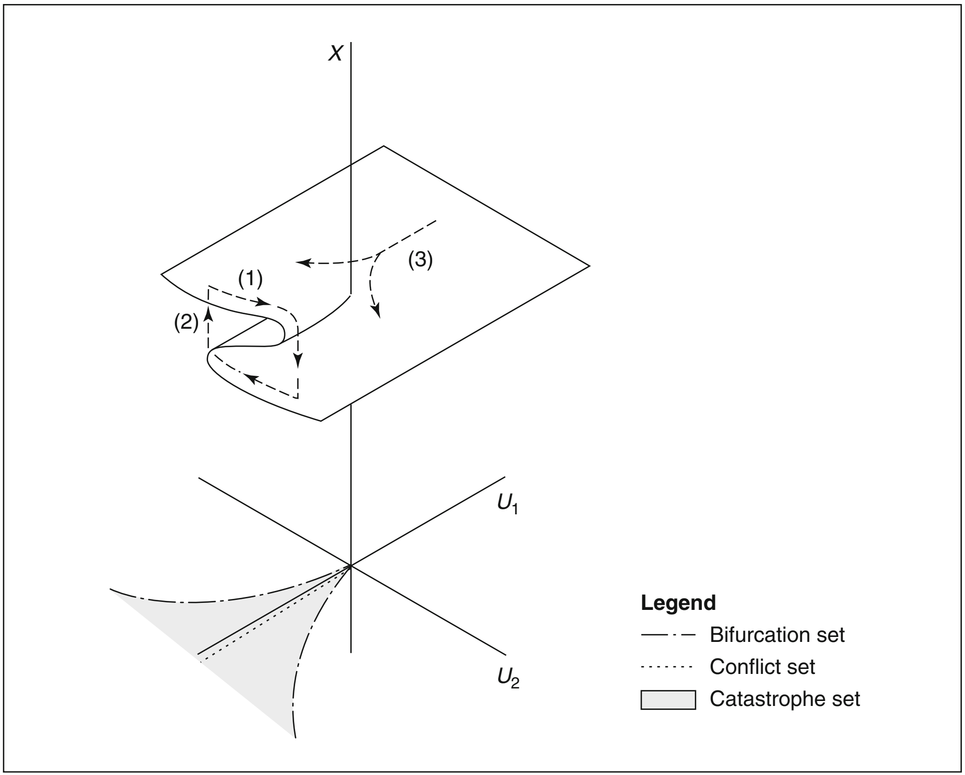
We can illustrate the possibilities of catastrophe theory using this figure. The surface of possible equilibrium values describes all possible states of the system. A particular behavior of the system is a trajectory on the surface. The study of such surfaces for particular systems, therefore, allows us to investigate possible types of behavior, and we know that the surface must *in a topological sense* be of the form shown in the figure. The italicized qualification is an important one in practice and should be emphasized. It means that the surface of possible equilibrium values for a system can be forced into the form of Figure A1.2 after some smooth transformation of the variables, where necessary. This is known as a standard, or canonical form. The achievement of the appropriate transformation in applied work is often likely to be a very difficult task, though insights can often be gained without it being carried through explicitly.

Three types of behavior that we are not accustomed to expect are shown in sample trajectories on Figure A1.2:

(1) a sudden jump (or catastrophe);

(2) **hysteresis**—a reverse path to some point not being the same as the starting point; and

Figure A1.2 THE CUSP SURFACE



SOURCE: Wilson(1981). Reprinted with permission.

(3) divergence—a small difference in approach toward, in this case, a cusp point, leads the system to the upper or lower surface and hence to a very different state.

It can easily be seen that the jump behavior arises from a path in the **U**-plane that leads the system to fall from the upper surface to the lower one at a fold (in other words, a change in state)—or vice versa.

It can also be seen that a fold, and hence jump behavior, arises because in some regions of **U**-space, there are multiple equilibrium solutions for **X**. In the particular case of Figure A1.2, there is a region in the central part of the diagram (at the fold) where there are three possible solution sets for **X**. It turns out that the upper and lower surfaces represent stable minima (and hence are observable) while the central part of the fold represents maxima and hence unstable (and unobservable) states. If this fold region is projected vertically downwards onto the **U**-plane, we obtain the familiar cusp-shaped section of that plane. This contains the set of values of **U** that are critical. Outside the shaded region, the system only has one state available to it; inside there are two possible observable states and hence possible conflict. As the boundary of the critical region is crossed, jumps can take place. We will see later in this section, and also in Section VI, how this preliminary analysis can be formalized into various concepts of stability. We will also see explicitly that, as noted above, the function *f* is singular at critical parameter values and that geometrically this can be identified with folds in the equilibrium surface. In the critical region, where there are multiple states, some rule has to be assumed, or discovered, about which state the system actually adopts. This involves a delay convention, which will be pursued later.

B. Elementary Catastrophes

We can examine the types of singularity in relation to canonical forms of functions and exploit the fact that other functions of the same co-rank[4] and co-dimension[5] can then be transformed (locally, in the neighborhood of a point) into the same form. In general, the canonical forms are polynomials consisting of a single state variable X, and assuming the form

$$f(X) = X^m + U_1 X^{m-2} + U_2 X^{m-3} + \ldots + U_{m-2} X \qquad \text{(A1.24)}$$

The first term, in this case X^m, captures the degeneracy and type of singularity. If all the **U**-variables are zero, this can be considered as the lowest order non-zero term in a Taylor expansion. As the **U**-variables vary from zero values, the right hand side of Equation A1.24 approximates the Taylor expansion of a whole family of functions. Catastrophe theory essentially says that all other families of functions with the same number of parameters have singularities of the same type as the canonical, truncated Taylor expansion. This form is said to represent a universal unfolding of singularities of this type. **Thom's theorem** says that for m up to six (that is, up to four control variables) this models all functions f of that co-dimension. It also models the structure of the singularities, in this neighborhood of the function. The canonical form can then be used as a model for the singularities of all the functions of this type.

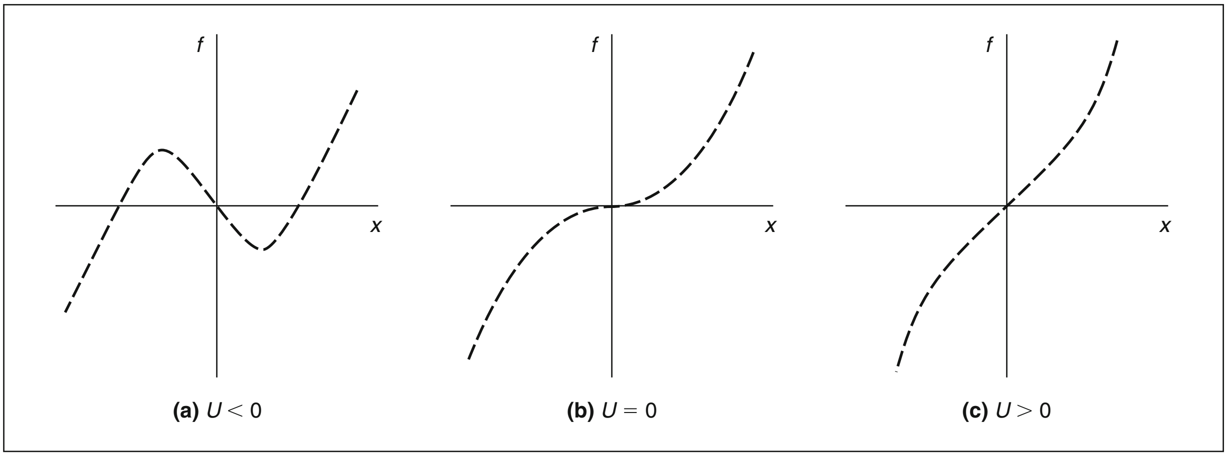
The notion of unfolding can also be expressed in another way, based on the concept of structural stability, which provides another route into catastrophe theory. Consider the function

$$f(X) = X^3 + UX \qquad \text{(A1.25)}$$

which is a special case of Equation A1.24 with $m = 3$. This is plotted in Figure A1.3 for the cases $U < 0$, $U = 0$, $U > 0$. When $U = 0$, $f(X) = X^3$ is not structurally stable in the sense that the addition (or substraction) of a term UX, however small U is, changes the shape of the curve in a basic way in the neighborhood of the origin. The function $f(X)$ in Equation A1.25 when $U \neq 0$, however, is structurally stable: it retains its shape under small perturbations. However, $f(X) = X^3$ is said to have a degenerate singularity at $X = 0$, and the addition of the term UX is the simplest way to make the function structurally stable.

We noted earlier that catastrophes occur because of the existence of multiple minima of the potential function. The behavior manifold is defined as the surface in (**X**, **U**) space that contains the minima of the potential function, the possible equilibrium states of the system. We can usefully classify different possible types of system behavior by focusing on the control manifold, the equilibrium surface in the smaller dimensional **U**-space. For each point on the control manifold, consider the point or points (if any) to which it gives rise on the behavior manifold. We can identify regions of the control manifold as follows:

Figure A1.3 ILLUSTRATING STRUCTURAL STABILITY



(a) $U < 0$ **(b)** $U = 0$ **(c)** $U > 0$

(1) The region where those values of the control variables generate only one equilibrium solution, and the behavior of the system is then well-determined.

(2) The region where there is more than one solution, discounting for the time being any region where there may be no solution. (This is known as the catastrophe set, and it is not immediately clear which state the system adopts, additional information must be supplied.)

(3) The bifurcation set, which is the set of points that separates the catastrophe set from the "single solution" set. (It is the critical set of points at which a minimum disappears. It is at such points that the system must jump to another state, and hence branch or bifurcate).

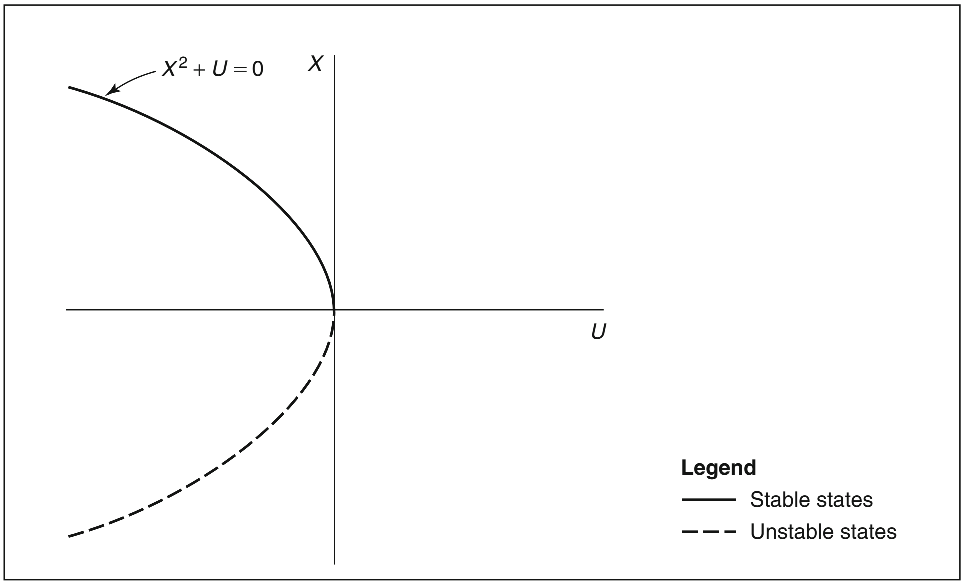
Notice these concepts have been illustrated in the cusp surface Figure A1.2. The precise behavior of the system for control points within the catastrophe set is determined by a delay convention as mentioned. This is a rule that must be supplied to determine which of the multiple possibilities the system adopts. The two most common are first, **perfect delay**, which means that the system stays in its original state until that state disappears as the trajectory leaves the bifurcation set; and second, the **Maxwell convention**, which assumes that if more than one minimum is available, the system chooses the state that represents the lowest. In the perfect delay case, jumps take place as the trajectory crosses the bifurcation line, as noted. In the Maxwell case, the region of interest is the so-called conflict set, defined as the points on the control manifold at which two or more minima take equal values. This has been illustrated in Figure A1.2 for the cusp case. With perfect delay, system behavior can be associated with thresholds that the system must cross before a change. In the case of the Maxwell convention, the conflict set can be seen as a traveling wave that is the basis for morphogenesis (or structural development), which is particularly important where the control variables are taken as representing space and time (three space coordinates and one time coordinate). It is in this context that the traveling wave concept plays a key role in applications.[6]

C. The Fold Catastrophe as an Example

The simplest of elementary catastrophes is the **fold**. It is the universal unfolding of the singularities of X^3 and its potential function is $f = X^3/3 + UX$ for a single state variable X and a single control variable U. The possible equilibrium states of this system are those for which f is a minimum, and we can find this by setting the derivative to zero: $df/dX = X^2 + U = 0$. This has solutions $X = \pm\sqrt{-U}$, and we note that the second derivative is $d^2f/dX^2 = 2X$. Since the derivative is positive for positive values of X and negative for negative values, the minima occur for the positive values and the maxima for negative values. The solution also shows that real roots only exist for negative U. This information is displayed in Figure A1.4, which shows a parabola. The top half has been shown as a solid curve, because it represents the minima and the stable, observable states of the system. The bottom half is in dashed lines; it represents the maxima, which are unstable and unobservable.

We can now illustrate the general argument in the previous subsection by this simple example. The function f is a canonical representation for any function with a singularity of co-rank 1 at the origin and of co-dimension 1. (In other words, $d^2f/dX = 2X$ is a first-order polynomial that vanishes at the origin, and there exists only one control variable.) In this case, since we have only one state variable and one control variable, the whole picture of possible equilibrium values—the singularities of f in the neighborhood of the origin, can be represented in two dimensions as shown in Figure A1.4. The control manifold, the projection of the (**X**, **U**)-manifold onto the **U**-manifold, is in this case simply the horizontal axis. There is no catastrophe set because there are no points on the horizontal axis at which there are two or more values of X for which f is a minimum. The bifurcation set is also very simple: it is the single point at the origin, because here an observable minimum disappears. It is at this point, therefore,

Figure A1.4 EXAMPLE OF A FOLD CATASTROPHE



SOURCE: Wilson(1981). Reprinted with permission.

that jump behavior can be observed. If the system is in a state given by negative U and on a trajectory in which U is increasing, then as U passes through zero, the stable minimum equilibrium-state disappears, and the system will have to take up some other state not accounted for by this diagram.

Because there is no area of the control manifold that produces multi-valued solutions, we cannot illustrate directly the concepts of delay conventions, conflict sets, and so on. However, we can see by reference to the fold catastrophe example in the book CD/DVD under the YiChan directory and in the "Activity Allocation and Derivation" chapter of Chan (2005) that even in this case, states can be added in the particular application that do create multi-valuedness. The example mentioned was concerned with the emergence or otherwise of spatial structure, namely whether or not to develop a housing project. Obviously delay conventions and associated concepts for thresholds are very important in this context.

One other technique can be introduced at this stage that gives more insight into the workings of catastrophe theory. Plots such as Figure A1.4 give the equilibrium values for X and U but do not show what is happening to the function f in the neighborhood of these values. This can be depicted on another form of diagram as illustrated in Figure A1.3: structural stability. For typical values of the control variable—in this case $U < 0$, $U = 0$ and $U > 0$—we can plot f against the state variable, in this case X. In the $U < 0$ case, the minimum (occurring at a positive value of X) can easily be seen as can the way in which the plot of f against X changes as U increases from a negative value. At $U = 0$, the graph is an obviously limiting case: the maximum and minimum have fused to form a point of inflexion, while for $U > 0$, the stationary points have clearly disappeared.

D. Higher Order Catastrophes

Aside from the fold and cusp catastrophes, there are other potential functions where catastrophes could occur. Of these, the seven elementary catastrophes are listed in Table A1.1. The table gives the number of state variables, the number of

Table A1.1 THE SEVEN ELEMENTARY CATASTROPHES

Name	State variables	Control variables/ Co-dimension	Potential function
Fold[a]	1	1	$X_1^3/3 + U_1X_1$
Cusp	1	2	$X_1^4/4 + U_1X_1^2/2 + U_2X_1$
Swallow tail[a]	1	3	$X_1^5/5 + U_1X_1^3 + 3 + U_2X_1^2/2 + U_3X_1$
Hyperbolic[a] umbilic	2	3	$X_1^3/3 + X_2^3/3 + U_1X_1X_2 - U_2X_1 - U_3X_2$
Elliptic[a] umbilic	2	3	$X_1^3/3 - X_1X_2^2/2 + U_1(X_1^2+X_2^2)/2 - U_2X_1 - U_3X_2$
Butterfly	1	4	$X_1^6/6 + U_1X_1^4/4 + U_2X_1^3/3 + U_3X_1^2/2 + U_4X_1$
Parabolic umbilic	2	4	$X_1^2X_2/2 + X_2^4/4 + U_1X_1^2/2 + U_2X_2^2/2 - U_3X_1 - U_4X_1$

[a] These unfolding functions are self-duals.

SOURCE: Wilson (1981). Reprinted with permission.

control variables, and the potential function that gives the universal unfolding of that type of singularity. Only the fold and cusp can be given a fully geometrical treatment because in other cases four or more dimensions would be needed for equivalent presentations. However, it is possible to generate pictures by portraying two or three dimensions as slices of higher order diagrams. The full treatment of the remaining elementary catastrophes are available in many other sources that can be used for reference (Wilson 1981).

The list of elementary catastrophes can be extended slightly by looking at duals. These exist for three of the entries in Table A1.1: the cusp, the butterfly and the parabolic umbilic; the rest of the list are self-duals. Duals are constructed by replacing the function being unfolded by its negative. In effect, this means that the positions of maxima and minima are reversed as the control manifold is covered. The self-duals are such because the negative sign can be produced by a change of coordinates. In effect, a review of Table A1.1 shows that functions that are wholly even-powered polynomials have duals which are different, while the rest, including at least one odd-powered term, are not, as replacing X by $-X$ produces the required minus sign. (For example, in the case of the fold, X^3 simply becomes $-X^3$ on this transformation.)

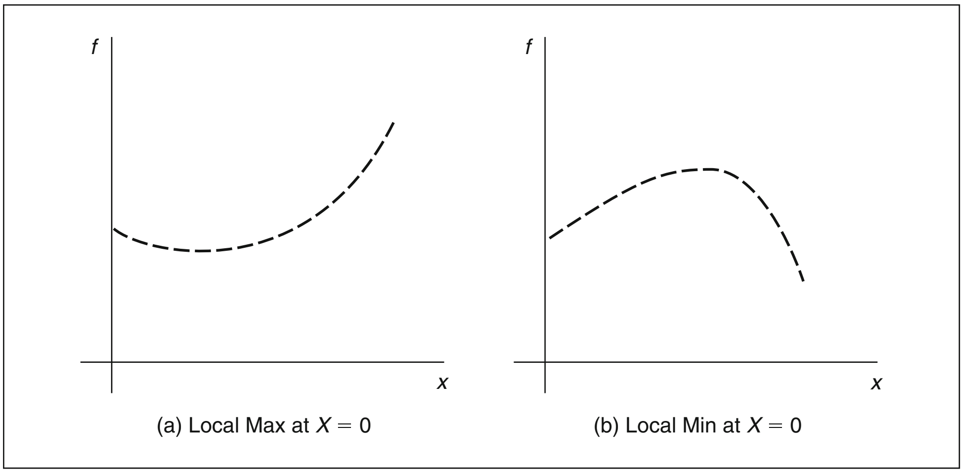
We illustrate briefly the concept of a dual by reference to the cusp. The potential function in Table A1.1 is replaced by $f = -X^4/4 + U_1 X^2 + U_2 X$ and the equivalent of Figure A1.4 can be used to see what happens when maxima are turned into minima and vice versa. The only maxima for the cusp surface were on the middle sheet of the folded section, and so this part of the surface in the dual becomes the only set of minima and therefore the only observable states. Thus there is a unique minimum inside the shaded areas of the control manifold and no stable states outside it. The possible behaviors of the system are therefore less interesting than that of the basic cusp surface. This particular example of catastrophe is also sometimes known as the false cusp.

Finally, we note the existence and importance of what is called **constraint catastrophes**. These arise as, in effect, extensions of Thom's theorem. The theorem is concerned with maxima and minima determined by points of the potential function where the derivative varnishes. If a model is constructed that includes constraints, then observable minima may be determined by the constraint rather than by vanishing derivatives. This point is illustrated by the curve shown in Figure A1.5, which shows the effect of a non-negativity constraint on a variable. In this case, local maxima or minima often occur on the boundary imposed by the constraint, and the derivative of the potential function does not vanish at that point.

E. Remarks

The reader will recall that the title of this appendix is "Control, Dynamics, and System Stability." While catastrophe theory contributes toward the subject of this chapter qualitatively, our focus is really on the more general discussion of system stability. Toward this goal, we will find that bifurcation theory is more quantitative in its applications. Indeed it is likely that sudden changes addressed by bifurcation theory are most important in applied work, inasmuch as most dynamic systems of interest are not gradient systems. In other words, the corresponding differential equation cannot be reduced to the optimization form $\nabla f(\mathbf{X}) = \mathbf{0}$. Typically, such differential equations have a small number of

Figure A1.5 LOCAL OPTIMA CREATED BY A CONSTRAINT



(a) Local Max at $X = 0$

(b) Local Min at $X = 0$

isolated equilibrium points, and information about system behavior is presented as trajectories on state-space diagrams. A continuous network example is shown in the "Chaos, Catastrophe, Bifurcation, and Disaggregation" chapter in Chan (2005). We will turn to these subjects sequentially in the sections below, starting with time trajectories on state space. Meanwhile, it is interesting to note that variational inequality may help identify the existence and uniqueness of equilibria.

V. COMPARTMENTAL MODELS

We have made a distinction throughout this book between prescriptive and descriptive analysis procedures. While the pevious sections of this appendix have dealt with prescriptive procedures, the techniques involved here are specific to the description of systems in terms of processes. Indeed, in this case, no variational principle, characteristic of the prescriptive approach, seems to apply. The models currently used can be subdivided in two classes: the deterministic models in terms of differential or difference equations and the stochastic models in terms of Markov processes or chains. More recently, quasi-deterministic models in terms of differential or difference stochastic equations have been developed. These concepts apply readily to compartmental models, subject of our present discussion (dePalma and Lefèvre 1987; Godfrey 1983; Seber and Wild 1989).

A. Basics

A compartmental model is concerned with the description of a system divided into a finite number of subsystems called compartments, between which the fundamental units of the system move. The purpose of this model is to describe the temporal evolution of the state of the system, which is defined as the number of units in the different compartments. The compartmental models defined here are intrinsically dynamic because the state of the system is the consequence of the various past transitions. The results derived describe the transient (finite time) and stationary (infinite time) regimes of the models. Moreover, the systems considered are concerned with characterizing the state variable in terms of populations or units, and special attention is paid to macroscopic behaviors and collective phenomena.

The most general form of compartmental equations for a system with n compartments is:

$$\frac{dX_i}{dt} = -H_{i0}' - \sum_{j \neq i} H_{ij}' + \sum_{j \neq i} H_{ji}' + H_{0i}' \qquad i = 1, \ldots, n \qquad \text{(A1.26)}$$

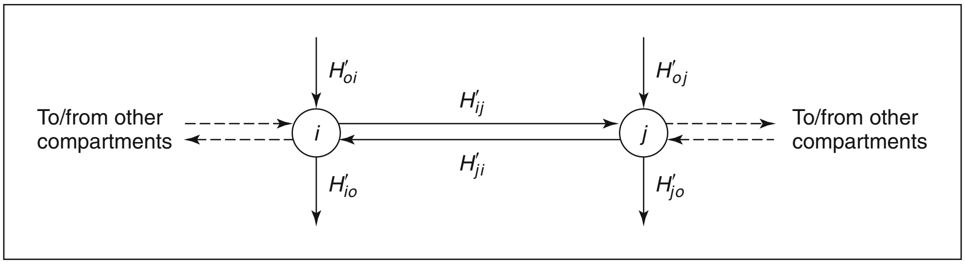
where X_i is the number of units in compartment i; H_{ij}' is the flow rate from compartment i to compartment j and the subscript 0 denotes the environment. Notice here that the flow rate H' is usually a function of the state variables, in other words, $H'(X_1, X_2, \ldots, X_n)$. If the flow rates from all compartments to the environment are zero ($H_{i0}' = 0, i = 1, \ldots, n$), the system is said to be closed; otherwise it is open. Equation A1.26 is illustrated for two of the n compartments in Figure A1.6.

Compartmental models typically involve rate constants like h in the growth model $dX(t)/dt = hX(t)$, where the growth rate is proportional to the population size $X(t)$ at time t. Thus $h = \frac{dX(t)/dt}{X(t)}$ or $dX(t) = hX(t)\,dt$. The reader can easily check using calculus that this single compartment model has solution $X(t) = X(0)\exp(ht)$. The example shows that compartmental models typically involve linear combinations of exponential terms, being solutions to differential equations. Consider the three-compartment model as shown in Figure A1.7, where an open system is portrayed, with both flow rates from and to the environment. Notice the input rate h_{01} from the environment is exemplified previously by the control variable U in our discussion of control theory. Here in this example, the change in the population in compartment 1 is $dX_1(t) = h_{21}X_2(t)\,dt + h_{01}dt - h_{13}X_3(t)\,dt - h_{12}X_1(t)\,dt$. Correspondingly, the rate of change is given by

$$\dot{X}_1 = \frac{dX_1(t)}{dt} = h_{21}X_2(t) + h_{01} - (h_{13} + h_{12})\,X_1(\,t) \qquad \text{(A1.27)}$$

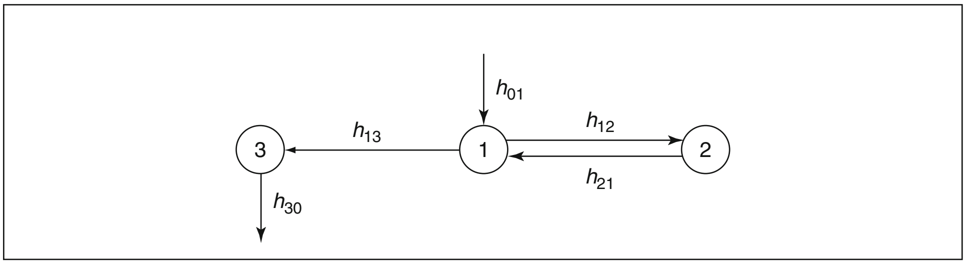
Thus the whole system of Figure A1.7 is described by the set of differential equations

Figure A1.6 ILLUSTRATING TWO COMPARTMENTS OF A GENERAL COMPARTMENTAL MODEL



SOURCE: Godfrey (1983). Reprinted with permission.

Figure A1.7 ILLUSTRATING A THREE-COMPARTMENT MODEL



SOURCE: Seber and Wild (1989). Reprinted with permission.

$$\begin{aligned}\dot{X}_1 &= (h_{13} + h_{12})X_1(t) + h_{21}\,X_2(t) + h_{01}\\ \dot{X}_2 &= -h_{12}X_1(t) - h_{21}X_2(t)\\ \dot{X}_3 &= h_{13}X_1(t) - h_{30}X_3(t)\end{aligned} \qquad \text{(A1.28)}$$

We see that compartmental models, in their fundamental form, are simply sets of constrained first-order differential equations, the constraints being the physical requirement that flow rates are non-negative. Two qualitatively different situations occur in this type of modeling. In the linear systems, the individuals have independent behaviors and consequently, the state of the population can be deduced simply from the behavior of their units. This applies to migration models, for example, where each migrant is supposed to make his/her decision independent of other migrants. In nonlinear models, the individuals have interdependent behaviors whose aggregation can give rise to qualitatively new situations. This applies to individual choice models in collective systems, where individual decisions are often made conditioned upon the state of the system. In other words, linear (time-invariant) compartmental models have flow rates that are directly proportional to the quantity in the donor compartment, with the constant of proportionality being referred to as a rate constant. Nonlinear systems, on the other hand, have some of the flow rates specified as a function of the state vector **X** instead of being constants.

B. Stochastic Models

While the above examples are illustrated for a deterministic case to fix ideas, the concept carries over readily to the stochastic case. For a stochastic compartmental model, we make here the Markov assumption that states that the individuals move from compartment to compartment with probabilities that depend on the characteristics of these compartments but not on the previously occupied compartments. In other words, it has the typical Markovian property that the future depends on the present but not on the past[7]. The situation is exactly analogous to deterministic models. In the linear case, the transition probabilities do not depend on the state of the system (that is to say, on the number of individuals in the compartment). In the nonlinear case, they do.

1. The Master Equation. Let us consider a compartment model with n compartments. $X_j(t)$ will denote the number of individuals in compartment j at time t, $t \geq 0$, and X_j^* a positive realization of $X_j(t)$, $j = 1, \ldots, n$. Let $\mathbf{X}(t)$ be the $n \times 1$ column vector $(X_1(t), \ldots, X_n(t))^T$ and $\mathbf{X}^*$ a possible realization of $\mathbf{X}(t)$. We define by $\pi_{ij}(\mathbf{X}^*, t)dt$, $1 \le i \ne j \le n$, the probability that during the infinitesimal time interval $(t, t + dt)$, a given individual moves from compartment i to compartment j when the system is in the state $\mathbf{X}^*$ at time t. Let $P(\mathbf{X}_0^*, \mathbf{X}^*, t)$ denote the probability that $\mathbf{X}(t) = \mathbf{X}^*$ given the initial condition $\mathbf{X}(0) = \mathbf{X}_0^*$. Let $\boldsymbol{\delta}_j = (\delta_{j1}, \ldots, \delta_{jr})$, $j = 1, \ldots, n$, be an orthonormal[8] base of the transition rate space II^n, which maps out the possible transitions from the current jth compartment.

The Kolmogorov forward differential equations[9] give the temporal evolution of the probability distribution of the system's state. This accounts for the gain terms allowing for transition to the state $\mathbf{X}^*$, and the loss terms allowing for transitions *from* the state $\mathbf{X}^*$. These equations take the following form where the gain terms are positive and the loss terms negative:

$$\begin{aligned}
\dot{P}(\mathbf{X}_0^*, \mathbf{X}^*, t) = & \sum_{i=1}^{n} \sum_{j \ne i} P(\mathbf{X}_0^*, \mathbf{X}^* + \boldsymbol{\delta}_i - \boldsymbol{\delta}_j, t)(X_i^* + 1)\pi_{ij}(\mathbf{X}^* + \boldsymbol{\delta}_i - \boldsymbol{\delta}_j, t) \\
& + \sum_{j=1}^{n} P(\mathbf{X}_0^*, \mathbf{X}^* + \boldsymbol{\delta}_j, t)(X_j^* + 1)\pi_{j0}(\mathbf{X}^* + \boldsymbol{\delta}_j, t) \\
& + \sum_{j=1}^{n} P(\mathbf{X}_0^*, \mathbf{X}^* - \boldsymbol{\delta}_j, t)\pi_{0j}(\mathbf{X}^* - \boldsymbol{\delta}_j, t) \qquad \text{(A1.29)} \\
& - P(\mathbf{X}_0^*, \mathbf{X}^*, t)\left[\sum_{i=1}^{n} \sum_{j \ne i} X_i^* \pi_{ij}(\mathbf{X}^*, t) \right. \\
& \left. + \sum_{j=1}^{n} X_j^* \pi_{j0}(\mathbf{X}^*, t) + \sum_{j=1}^{n} \pi_{0j}(\mathbf{X}^*, t) \right]
\end{aligned}$$

The above constitute the master equations for the multvariate birth-and-death process. The notations used are consistent with those in Appendix 3. These equations are difficult to solve in general. Short of a solution, however, there are means to obtain information on the evolution of the system's state, as we will demonstrate.

2. A Special Nonlinear Case. In the nonlinear case, the transition rates $\pi_{ij}(\mathbf{X}^*, t)$, $\pi_{j0}(\mathbf{X}^*, t)$, $\pi_{0j}(\mathbf{X}^*, t)$ depend explicitly on the state of the system. The presence of the argument $\mathbf{X}^*$ in the transition rates makes Equation A1.29 even more difficult to solve. However, we will discuss two examples that permit the derivation of analytical results. Let us consider a population of N individuals, each individual having to select one between two choices (1 and 2). The choice behavior of the individuals is described by a compartmental model, each choice corresponding to a compartment in the system. Assuming the system was initially empty, we will describe the state of the system with one of the two variables, $X_i(t)$, whose realizations are denoted by $\mathbf{X}^*$. Let us now give the structure of the transition rate $\pi_{ij}(\mathbf{X}^*, t)$, $1 \le i, j \le 2$.

We will suppose that an individual decides to review and modify the choice through two successive steps. First, during $(t, t + dt)$, an individual reviews his or her present choice i with a probability $\pi^{(i)}dt$; then, he or she selects a choice j with a probability $p^{(j)}(\mathbf{X}^*)$ which has the following *logit* form:[10]

$$p^{(j)}(\mathbf{X}^*) = \frac{\exp\,[v^{(j)}(\mathbf{X}^*)/\mu^{(j)}]}{\exp\,[v^{(1)}(\mathbf{X}^*)/\mu^{(1)}] + \exp\,[v^{(2)}(\mathbf{X}^*)/\mu^{(2)}]} \quad (j = 1, 2) \qquad \text{(A1.30)}$$

where $v^{(1)}(\mathbf{X}^*)$ and $v^{(2)}(\mathbf{X}^*)$ represent the utility functions of choices 1 and 2 respectively, and μ is a positive parameter tht expresses the degree of uncertainty in the individual's behavior. Specifically, we add an error term $\epsilon^{(j)}$ to the deterministic value function $v^{(j)}$: $v^{(j)}(\mathbf{X}^*) + \mu^{(j)}\epsilon^{(j)}$, where $\mu^{(j)}$ is a constant measuring the importance of the error term. Thus $\mu^{(j)}$ can be thought of as a scaling constant; the larger it is, the higher the uncertainty.

It is clear that the process for determining $X_i(t)$ is reduced to a birth-death process[11]. The global distribution of individual choices strongly depends on the structure of the utility functions. For illustration, we will examine the cases where the utilities of each choice is a linear or logarithmic function of the number of individuals who have adopted this choice (that is, $v^{(1)}(\mathbf{X}^*)$ and $v^{(2)}(\mathbf{X}^*)$ are linear or logarithmic functions of X^* and $N–X^*$ respectively).

Let us first consider the case where the utility functions are linear:

$$\begin{aligned} v^{(1)}(\mathbf{X}^*) &= a + bX^* \\ v^{(2)}(\mathbf{X}^*) &= c + d(N - X^*) \end{aligned} \quad X^* = 0, \ldots, N \tag{A1.31}$$

The Markov process is then irreducible (that is to say, all compartments intercommunicate), $\lim_{t\to\infty} P(X_0^*, X^*, t) \equiv P(X^*)$ exist and are independent of the initial condition X_0^*. Define the generating function[12] $G(\xi, t) = \sum_{\mathbf{X}^*} \xi^{\mathbf{X}^*} P(\mathbf{X}_0^*, \mathbf{X}^*, t)$ where the nth-derivative exists for $G(\boldsymbol{\xi}, t)$ $|\xi_j| < 1, 1 \leq j \leq n; \xi^{\mathbf{x}^*} \equiv \xi_1^{\mathbf{x}_1^*}, \ldots, \xi_n^{\mathbf{x}_n^*}$. The generating function for the probability distribution $P(\mathbf{X}_0^*, \mathbf{X}^*, t)$, where $\mathbf{X}^* = [X_1^*(t), X_2^*(t), \ldots, X_n^*(t)]^T$, can be written out in long hand for a stationary, irreducible Markov process. It assumes the form $P(\mathbf{X}_0^*) + \xi_1^{X1^*}P(X_1^*) + \xi_2^{X2^*}P(X_2^*) + \cdots + \xi_n^{Xn^*}P(X_n^*)$. Suppose there are no arrivals and departures. Then for the stationary solution $\lim_{t\to\infty} G(\xi, t) = (\mathbf{p}^T\xi)^{X_0^T\mathbf{u}''}$ where $\mathbf{p}$ is a $n \times 1$ Perron vector whose components are positive and of sum equal to 1, and $\mathbf{u}''$ is an $N \times 1$ column vector $(1, \ldots, 1)^{T\cdot}$ Consequently, at the stationary state, G is the generating function of a multinomial vector of exponent $\mathbf{X}_0^T\mathbf{u}''$ and of parameter $\mathbf{p}$ (Cox and Miller 1965). In long hand,

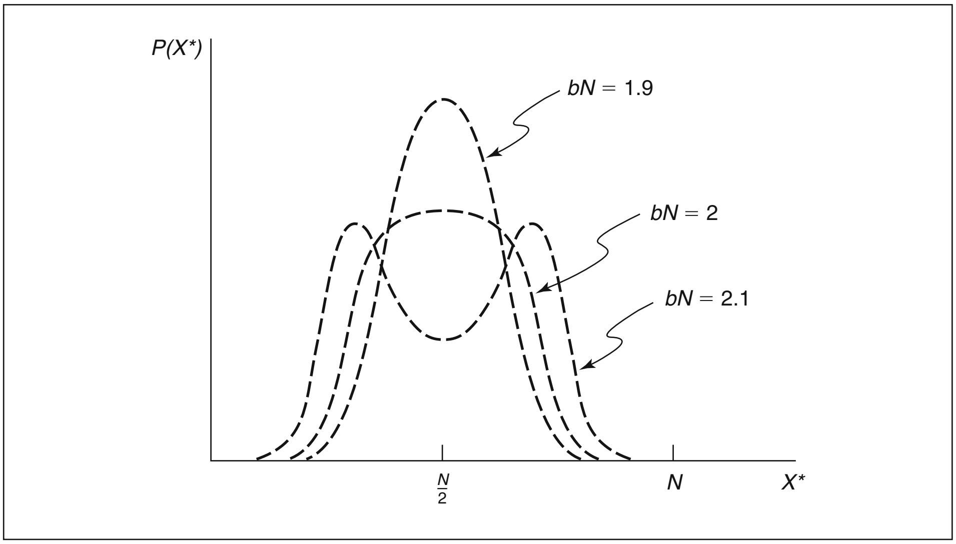
$$G(\boldsymbol{\xi}, t) = [p^{(1)}\xi_1^{X1^*} + p^{(2)}\xi_2^{X2^*} + \ldots + p^{(\mathrm{n})}\xi_n^{Xn^*}]^{\mathrm{constant}}$$

We will now examine the symmetrical situation where $\pi^{(1)} = \pi^{(2)}$, $a = c$, and $b = d$ to illustrate in a simple way the importance of nonlinearities. Clearly, the stationary distribution is symmetrical, and $\mu^{(1)} = \mu^{(2)} = \mu$. When $b = 0$ (linear case), the stationary state can be shown via the above generating function result to be a binomial distribution of exponent N and parameter 0.5,

$$P(X^*) = \binom{N}{X^*}[\exp(-0.5)]^{X^*}[1 - \exp(-0.5)]^{N - X^*}$$

The values of b tht are positive (negative) express a behavior of imitation (antilimitation). It can be proved that if the imitation behavior becomes sufficiently important (or when $b > 2\mu/\mathrm{N}$), the stationary distribution passes from a unimodal to a bimodal shape: the state $N/2$ is no longer the mode of the distribution and corresponds now to a local minimum of the distribution. This is illustrated in Figure A1.8.

Figure A1.8 STATIONARY DISTRIBUTION IN THE NONLINEAR CASE WHEN $\mu = 1$



SOURCE: dePalma and Lefèvre (1987). Reprinted with permission.

Let us now consider the case where the utility functions are logarithmic:

$$\begin{matrix} v^{(1)}(X^*) = a + b \ln X^* \\ v^{(2)}(X^*) = c + d \ln N - X^* \end{matrix} \quad X^* = 0, \ldots, N \tag{A1.32}$$

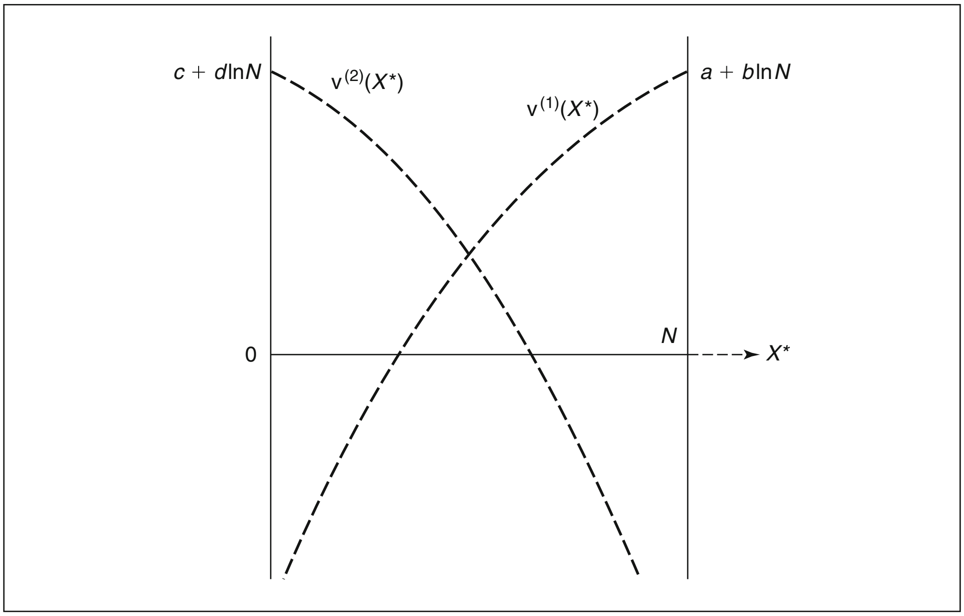
The Markov process is then either irreducible or absorbing in relation to the sign of the coefficients b and d. For example, when b and d are positive, there exists two absorbing states 0 and N. In other words, the transitional probability $\pi_{00} = 1$ and $\pi_{NN} = 1$. In this case, it is quite plausible that one of these two states, 0 for example, is in fact preferable to the other, N. As the absorption probabilities depend on μ, it is then natural to consider this parameter (interpreted here as the information level accessible to individuals) as a control parameter to maximize the probability of absorption in state 0. It can be proved that here exists an optimal stationary policy that consists of taking for μ the largest possible value when the choice distribution X^* favors choice 1 to the detriment of choice 2 (that is when $v^{(2)}(X^*) < v^{(1)}(X^*)$) and the smallest possible in the contrary case. This is illustrated in Figure A1.9.

The above two models represent examples of an epidemic model. Such ecological models are related to the Lotka-Volterra predator-prey model as well as nonlinear, dynamic, Lowry-derivative models[13]. They typically describe the interacting (often conflicting) relationship between two or more populations. More importantly, they illustrate the asymptotic behavior of stochastic models. Often, a stationary solution is obtained that can be adequately modeled by a deterministic framework.

C. Deterministic Models

A deterministic version of Equation A1.29 can be written in terms of the following differential equations:

Figure A1.9 LOGARITHMIC UTILITY FUNCTION



SOURCE: dePalma and Lefèvre (1987). Reprinted with permission.

$$\begin{aligned}\frac{dX_i(t)}{dt} &= \sum_{j\neq i} X_j(t)H'_{ji}[\mathbf{X}(t), t] - X_i(t)\sum_{j\neq i} H'_{ij}[\mathbf{X}(t), t] \\ &\quad -X_i(t)H'_{i0}[\mathbf{X}(t), t] + H'_{0i}\,[\mathbf{X}(t), t] \qquad i = 1, \ldots, n\end{aligned} \tag{A1.33}$$

where $\mathbf{X}(t) \cong [X_1(t), \ldots, X_n(t)]^T$ is the $n \times 1$ column vector of the state of the system at time t, and $H'_{ij}(\mathbf{X}, t)$'s represent the deterministic flow rates defined before. In the linear case, the systems of differential equations can be written in the form

$$\dot{\mathbf{X}}(\mathbf{t}) = \mathbf{A'X(t)} + \mathbf{U(t)} \tag{A1.34}$$

subject to $\mathbf{X}(0) = \mathbf{X}_0$. Here $\mathbf{U}(t)$ can refer to an input vector or control variables. Such equations are discussed in many books on differential equations and dynamical systems. They involve the use of matrix exponential exp $(\mathbf{A}')$ for a square matrix $\mathbf{A}'$. This is defined as $\exp(\mathbf{A}') = \mathbf{I}+\mathbf{A}'+\mathbf{A}'^2/2! + \mathbf{A}'^3/3! + \cdots$ and this series converges for any $\mathbf{A}'$. If $\mathbf{A}'$ is any square matrix, the general solution of the homogeneous[14] equation $\dot{\mathbf{X}} = \mathbf{A}'\mathbf{X}$ is given by $\mathbf{X} = \exp\,(\mathbf{A}'t)\mathbf{c}$ for any constant vector $\mathbf{c}$. A particular solution[15] is $\int_0^t \exp\,(\mathbf{A}'(t - \tau))\mathbf{U}(\tau)\, d\tau$. Thus the complete solution to Equation A1.34 that satisfies the initial conditions is

$$\mathbf{X}(t) = \exp\,(\mathbf{A}'t)\mathbf{N}_0 + \int_0^t \exp\,(\mathbf{A}'\,(t - \tau))\mathbf{U}(\tau)\,\mathrm{d}\tau \tag{A1.35}$$

When the $n \times n$ matrix $\mathbf{A}'$ has n linearly independent eigenvectors[16], it is possible to form the spectral decomposition $\mathbf{A}' = \mathbf{P}''\mathbf{Q}''\mathbf{P}'^{-1}$, where $\mathbf{Q}''$ is an $n \times n$ matrix with n eigenvalues on its diagonal, or $\mathbf{Q}'' = \text{diag}\,(q_1{}', q_2{}', \ldots, q_n{}')$, and the kth column of $\mathbf{P}'$ is a right eigenvector of $\mathbf{A}'$ corresponding to $q_k{}'$. In particular, this is possible if all the eigenvalues of $\mathbf{A}'$ are distinct, as this implies that the eigenvector are all linearly independent.

In the linear case the state of the system in the deterministic version is the expected state in the Markovian version for identical initial conditions. In the nonlinear case, this result is no longer true. The Kurtz theorem establishes a connection, under certain hypotheses, between the deterministic and Markovian models. This result can be applied when the total population is important, say proportional to a large number N'(large). In addition, it supposes that the arrival rates in the system take the form $N'(\text{large})H'_{0i}$, and that the rates, H'_{0i}, H'_{i0} and H'_{ij} $(1 \le i, j \le n)$, depend on the state of the system through the relative frequencies $\mathbf{X}/N'$(large) (and not on the absolute values $\mathbf{X}$).

Let $\mathbf{Z}(t)$[large] denote the relative frequency (or density for short) $\mathbf{X}/N'$(large) and $\mathbf{Z}^*(t)$ the density $\mathbf{X}^*(t)/N'$(large) for the stochastic and deterministic versions respectively. Kurtz has proved the following theorem: Under the hypothesis given above, if $\lim_{N'(\text{large}) \to \infty} \mathbf{Z}(0)[N'(\text{large})] = \mathbf{Z}^*(0)$, then for every τ $(0 \le \tau \le \infty)$ $\lim_{N'(\text{large}) \to \infty} \sup_{t \le \tau} \mid \mathbf{Z}^*(t[N'(\text{large})] - \mathbf{Z}^*(t) \mid = \mathbf{0}$. That is to say, the normalized state in the Markovian version converges almost always to the normalized state in the deterministic version. This allows one to use the deterministic model to approximate the stochastic.

D. Deterministic Example

Consider an example as illustrated in Figure A1.10, where the constant transition rates h are shown. In this open system, there is an initial quantity of $N_1^*(0)$ in compartment 1 and nothing elsewhere. The rate of change equations can be written as

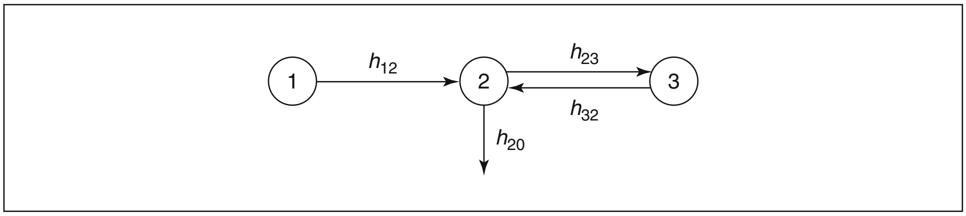
$$\begin{aligned} \dot{N}_1 &= -h_{12}N_1 \\ \dot{N}_2 &= h_{12} + h_{32}N_3 - (h_{23} + h_{20})N_2 \\ \dot{N}_3 &= h_{23} - h_{32}N_3 \end{aligned} \tag{A1.36}$$

with initial conditions $N_1^*(1, 0, 0)^T$ and the parameters $\{N_1^*, h_{12}, h_{23}, h_{32}, h_{20}\}^T$. Hence

$$\mathbf{A}' = \begin{bmatrix} -h_{12} & 0 & 0 \\ h_{12} & -(h_{23} + h_{20}) & h_{32} \\ 0 & h_{23} & -h_{32} \end{bmatrix}$$

$$\mathbf{A}' - q'\mathbf{I} = \begin{bmatrix} -h_{12} - q' & 0 & 0 \\ h_{12} & -h_{23} - h_{20} - q' & h_{32} \\ 0 & h_{23} & -h_{32} - q' \end{bmatrix}$$

Figure A1.10 EXAMPLE OF A DETERMINISTIC COMPARTMENTAL MODEL



SOURCE: Seber and Wild (1989). Reprinted with permission.

The characteristic polynomial[17] for the eigenvalues q' is

$$|\mathbf{A}' - q'\mathbf{I}| = -(h_{12} - q')[q'^2 + (h_{23} + h_{32} + h_{20})q' + h_{32}h_{20}] = 0 \quad \text{(A1.37)}$$

with eigenvalues q' equal to $-h_{12}$ and $-1/2\{h_{23} + h_{32} + h_{20} \pm [(h_{23} + h_{32} + h_{20})^2 - 4\, h_{32}h_{20}]^{1/2}\}$. For simplicity, we will write the last two eigenvalues (out of three) as a and b, where $a + b = -(h_{23} + h_{32} + h_{20})$ and $ab = h_{32}h_{20}$.

The adjoint matrix is obtainable by replacing each element of a square matrix by its co-factor[18] and then interchanging rows and columns:

$$\text{adj}\,(\mathbf{A}' - q'\mathbf{I}) = \begin{bmatrix} (h_{23}+h_{20}+q')(h_{32}+q') - h_{23}h_{32} & 0 & 0 \\ h_{12}(h_{32} + q') & (h_{12}+q')(h_{32}+q') & h_{32}(h_{12}+q') \\ h_{12}h_{23} & h_{23}(h_{12}+q') & (h_{12}+q')(h_{12}+h_{20}+q') \end{bmatrix} \quad \text{(A1.38)}$$

It can be shown that *any* non-zero column of adj $(\mathbf{A}' - q'\mathbf{I})$ is a right eigenvector $\mathbf{x}_R$ of $\mathbf{A}'$ in the set of homogeneous equations $\mathbf{A}'\mathbf{x}_R = q'\mathbf{x}_R$. Similarly, *any* row is a left eigenvector $\mathbf{x}_L$, or $\mathbf{x}_L^T\mathbf{A}' = \mathbf{x}_L q'$. Thus the right eigenvectors corresponding to eigenvalues $-h_{12}$, a and b respectively (after some slight manipulation and canceling common column factors), are

$$\mathbf{R}' = \begin{bmatrix} | & | & | \\ \mathbf{x}(-h_{12}) & \mathbf{x}(a) & \mathbf{x}(b) \\ \downarrow & \downarrow & \downarrow \end{bmatrix} = \begin{bmatrix} (h_{12} + a)(h_{12} + b) & 0 & 0 \\ h_{12}(h_{32} + h_{12}) & h_{32} + a & h_{32} + b \\ h_{12}h_{23} & h_{23} & h_{23} \end{bmatrix} \quad \text{(A1.39)}$$

The left (row) eigenvectors using rows 1 and 3 of the adjoint matrix yield

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ h_{12}h_{23} & h_{23}(h_{12} + a) & (h_{12} + a)(h_{23} + h_{20} + a) \\ h_{12}h_{23} & h_{23}h_{12} + b) & (h_{12} + b)(h_{23} + h_{20} + b) \end{bmatrix} \quad \text{(A1.40)}$$

From linear algebra (Noble 1969), if $\mathbf{A}'$ has n independent right eigenvectors $\mathbf{x}_R$, then to each $\mathbf{x}_R$ there corresponds a left eigenvector $\mathbf{x}_L$ for the same eigenvalue such that $\mathbf{x}_L^T\mathbf{x}_R = 1$. We can thus normalize the rows of $\mathbf{L}$ using

$$\mathbf{x}(q') = \mathbf{x}_L(q')/\mathbf{x}_L(q')^T\,\mathbf{x}_R(q') \quad \text{(A1.41)}$$

The resulting vector $\mathbf{x}(q')$ is a row vector of $\mathbf{R}'^{-1}$ since $\mathbf{R}'^{-1}\mathbf{R}' = \mathbf{I}$:

$$\mathbf{R}'^{-1} = \begin{bmatrix} \dfrac{1}{(h_{12} + a)(h_{12} + b)} & 0 & 0 \\ \dfrac{h_{12}}{(h_{12} + a)(a - b)} & \dfrac{1}{a - b} & \dfrac{h_{23} + h_{20} + a}{h_{23}(a - b)} \\ \dfrac{h_{12}}{(h_{12} + b)(b - a)} & \dfrac{1}{b - a} & \dfrac{h_{23} + h_{20} + b}{h_{23}(b - a)} \end{bmatrix} \quad \text{(A1.42)}$$

The solution to homogeneous equations $\dot{\mathbf{X}} = \mathbf{A}'\mathbf{X}$ with initial condition $\mathbf{X}(0) = \mathbf{X}^*(0)$ can be written as $\mathbf{X}(t) = \exp(\mathbf{A}'t)\mathbf{X}^*(0) = \sum_{j=1}^{q} \beta_j \exp(q_j't)$ where $\beta_j = [\mathbf{x}(q')^T \mathbf{X}^*(0)]\, \mathbf{x}_R(q')$ and q is the number of eigenvalues. This has been referred to as the sum of exponential model, and is typical of linear systems. Given $\mathbf{X}(0) = N_1^*(0)(1, 0, 0)^T$, the solution $\mathbf{X}(t)$ in this case is

$$\mathbf{X}(t) = N_1^*(0) \sum_{j=1}^{3} \exp\,(q_j'\,t) \left(\mathbf{x}_R(q_j') \right) \left[\left(-\mathbf{x}(q_j') \rightarrow \right) \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \right] = N_{1(0)}^{*} \begin{pmatrix} w'_1 \\ w'_2 \\ w'_3 \end{pmatrix} \quad \text{(A1.43)}$$

Here,

$$w'_1 = \exp\,(-h_{12}t)$$

$$w'_2 = h_{12} \left[\frac{h_{32} - h_{12}}{(h_{12} + a)(h_{12} + b)} \exp\,(-h_{12}t) + \frac{h_{32} + a}{(h_{12} + a)(a - b)} \exp\,(at) \right.$$
$$\left. + \frac{h_{32} + b}{(h_{12} + b)(b - a)} \exp\,(bt) \right] \quad \text{(A1.44)}$$

$$w'_3 = h_{12}h_{23} \left[\frac{1}{(h_{12} + b)(a + b)} \exp\,(-h_{12}t) + \frac{1}{(h_{12} + b)(a - b)} \exp\,(at) \right.$$
$$\left. + \frac{1}{(h_{12} + b)(b - a)} \exp\,(bt) \right]$$

We note that $X_1(t) \overset{-}{=} N_1^*(0) \exp\,(-h_{12}t)$ corresponds to a simple exponential decay.

E. Stochastic Example

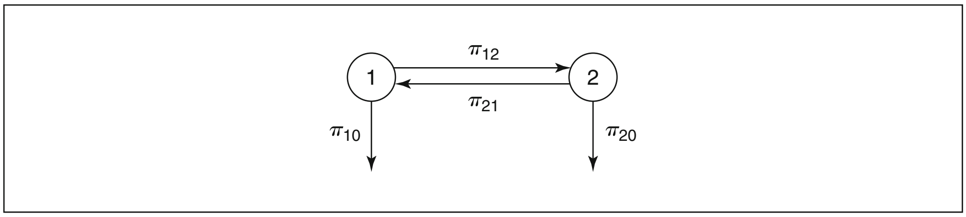
Shown in Figure A1.11 is a two-compartment open model. For this model, the parameters $\{\pi_{12}, \pi_{21}, \pi_{10}, \pi_{20}\}$ are given. Notice the flow rates are now denoted by π's instead of h's to show the stochastic nature of the current model, consonant with the notation used in Equation A1.29. In lieu of the rate-of-change matrix $\mathbf{A}'$, we write its stochastic counterpart as $\mathbf{II} = [\pi_{ij}]$. Here the matrix takes on the form

$$\mathbf{II} = \begin{bmatrix} 0 & \pi_{10} & \pi_{20} \\ 0 & -\pi_{10} - \pi_{12} & \pi_{21} \\ 0 & \pi_{12} & -\pi_{20} - \pi_{21} \end{bmatrix}$$

where the first row and column refer to transitions to and from the environment. For this model the explicit form for $\mathbf{P}(t) = \exp\,(\mathrm{II}t)$ is readily obtainable. The eigenvalues of $\mathbf{II}$ are $q_0' = 0$ and $q_1', q_2' = -1/2\{\pi_{10} + \pi_{12} + \pi_{20} + \pi_{21} \pm [(\pi_{10} + \pi_{12} - \pi_{20} - \pi_{21})^2 + 4\pi_{21}\pi_{12}]^{1/2}\}$. Now

$$\mathbf{P}(t) = \begin{bmatrix} 1 & p_{01}(t) & p_{02}(t) \\ 0 & p_{11}(t) & p_{12}(t) \\ 0 & p_{21}(t) & p_{22}(t) \end{bmatrix}$$

Figure A1.11 EXAMPLE OF A STOCHASTIC COMPARTMENTAL MODEL



SOURCE: Seber and Wild(1989). Reprinted with permission.

and using the method of the deterministic example above, including normalization via Equation A1.41, we find that

$$p_{11}(t) = \frac{1}{q'_2 - q'_1}\left[(\pi_{01} + \pi_{21} + q'_2) \exp(q'_1 t) - (\pi_{01} + \pi_{21} + q'_1) \exp(-q'_2 t)\right]$$

and (A1.45)

$$p_{21}(t) = \frac{\pi_{21}}{q'_2 - q'_1}\left[\exp(q'_2 t) - \exp(q'_1 t)\right]$$

The terms $p_{22}(t)$ and $p_{21}(t)$ are obtained by symmetry, and the values of $p_{01}(t)$ and $p_{02}(t)$ follow from the fact that the column sums of **P** are unity.

F. Discrete Time Models

Let us revisit a compartmental system where a discrete time scale $t = 0, 1, 2, \ldots$ is used instead of a continuous time axis. First, let us consider the stochastic model. We are interested in the linear case where the transition rates $\pi_{ij}(\mathbf{X}^*, t)$, $\pi_{0j}(\mathbf{X}^*, t)$ and $\pi_{j0}(\mathbf{X}^*, t)$ are independent of $\mathbf{X}^*$. If $p_{ij}(t)$ denote the probability that an individual in compartment i at time 0 is in compartment j at time t, $\mathbf{P}(t)$ is the $n \times n$ matrix of these $p_{ij}(t)$'s as mentioned. Equation A1.29 can be written in a compact form as $\mathbf{P}(t + 1) = \mathbf{P}(t)\mathbf{II}(t)$ where $\mathbf{II}(t)$ is the $n \times n$ matrix of the $\pi_{ij}(t)$'s. We note that if $\mathbf{II}(t)$ is a constant matrix **II**, then

$$\mathbf{P}(t) = \mathbf{II}^t \qquad \text{(A1.46)}$$

It can be shown through the use of generating functions that expected values of **X** can be written asymptotically ($t \to \infty$) as the difference equation

$$E[\mathbf{X}(t + 1)] = \mathbf{II}'E[\mathbf{X}(t)] + \mathbf{II}_0(t) \qquad \text{(A1.47)}$$

where $\mathbf{II}_0(t) = [\pi_{10}(\mathrm{t}), \ldots, \pi_{n0}(t)]^T$.

The deterministic version associated with the Markov model above is constructed formally by putting $\mathbf{X}^*(t + 1) \cong E[\mathbf{X}(t + 1) \mid \mathbf{X}(t)]$ and $\mathbf{X}^*(t) \equiv \mathbf{X}(t)$. Thus $\mathbf{X}^*(t)$ is solution of the following system of difference equations:

$$\mathbf{X}^*(t + 1) = \mathbf{A}'^t[\mathbf{X}(t), t]\mathbf{X}^*(t) + \mathbf{A}_0[\mathbf{X}(t), t] \qquad \text{(A1.48)}$$

where we replace **II** with $\mathbf{A}'$. In the linear case, the above equation reduces to Equation A1.47. This says that, as in the continuous time version, the state of the deterministic model is asymptotically equal to the expected state of the Markovian model (for identical initial conditions).

We have demonstrated above and in Section V-C that the normalized Markovian process converges to its associated deterministic version almost surely as the population size becomes very large. In applications, it is then natural to approximate the stochastic model by the deterministic one, in view of the computational advantage. However, the Kurtz theorem does not give any information on the quality of this approximation over time. It is possible to express the stochastic process as the sum of the deterministic process and a stochastic diffusion (epidemic) process[19]. The result allows us to judge the validity of the approximation as a function of time. Moreover, it is also very useful for statistical inference because a likelihood function can then be easily constructed from the data. We call this procedure the quasi-deterministic approach. While we will show an example, the reader is referred to dePalma and Lefèvre (1987) for details of this procedure.

Comparing the deterministic with the probabilistic evolution, Haag (1989) notes that the latter is the more general formulation, since the case of incomplete knowledge about the system comprises complete knowledge as a limit case while the converse is not true. The limit case of almost complete knowledge of the dynamics is revealed by the shape of the probability distribution $P(\mathbf{X}_0^*, \mathbf{X}^*, t)$ itself in the master equation A1.29. In this case, the master equation leads to an evolution in such a way that it develops one outstanding mode sharply peaked around the most likely state $\mathbf{X}^*_{\text{Max}}(t)$. This means that the system assumes state $\mathbf{X}^* = \mathbf{X}^*_{\text{Max}}(t)$ with overwhelming probability at time t, while all the other states $\mathbf{X}^* \neq \mathbf{X}^*_{\text{Max}}$ are highly improbable at the same time. Evidently this particular case descries a quasi-deterministic evolution of the system along path $\mathbf{X}^*(t) \approx \mathbf{X}^*_{\text{Max}}(t)$.

G. Example of a Quasi-Deterministic Analysis

Consider a closed system solution to the linear version of difference Equation A1.48 in which $\mathbf{A}_0 = \mathbf{B}_0 = \mathbf{0}$ and Equation A1.48 represents the individual terms of the geometric series[20]

$$\mathbf{I} + \tilde{\mathbf{B}} + \tilde{\mathbf{B}}^2 + \cdots + \tilde{\mathbf{B}}^k \qquad k \to \infty \tag{A1.49}$$

It can be shown that this deterministic approximation of the stochastic Equation A1.47 may or may not converge under certain circumstances (Yi 1986; Noble 1969). Here we will investigate the circumstances where convergence is guaranteed. Through this exercise, we wish to arrive at a stationary solution to Equation A1.49 and in the process show an example of how a seemingly complex stochastic model can be approximated asymptotically by a simple deterministic model.

Before we start, several basic concepts in matrix algebra need to be reviewed. Matrix norm on square matrix $\tilde{\mathbf{B}}$ is defined as $\|\tilde{\mathbf{B}}\| = \text{Max}_{|\mathbf{z}|=1} \|\tilde{\mathbf{B}}\mathbf{z}\|$ where $\mathbf{z} = \mathbf{x}/\|\mathbf{x}\|$, $\mathbf{x}$ is any vector, and $\|\mathbf{z}\| = 1$. In this spirit, the matrix norm $\|\tilde{\mathbf{B}}\|$ is parallel to the concept of a vector norm $\|\mathbf{x}\|$. On the other hand, the spectral radius of matrix $\tilde{\mathbf{B}}$, $\breve{\rho}(\tilde{\mathbf{B}})$, is defined as $\text{Max}_k\ |q_k'|$ where q_k's are eigenvalues of $\tilde{\mathbf{B}}$. For example, let $\tilde{\mathbf{B}} = \left[\begin{smallmatrix} 1 & c \\ 0 & 1 \end{smallmatrix}\right]$ for any real c. $\|\tilde{\mathbf{B}}\| = [1/2c^2 + 1 + 1/2c(c^2 + 4)^{1/2}]^{1/2}$ and the spectral radius $\breve{\rho}(\tilde{\mathbf{B}}) = 1$.

Now according to the definition of matrix norm, we can state for the ∞-norm

$$\|\tilde{\mathbf{B}}\|_\infty = \underset{\|\mathbf{z}\| = 1}{\text{Max}} \|\tilde{\mathbf{B}}\mathbf{z}\|_\infty \geq \|\tilde{\mathbf{B}}\mathbf{x}_i\|_\infty \qquad \text{(A1.50)}$$

where $\mathbf{x}_i$s are any normalized eigenvector (in other words, $\|\mathbf{x}_i\|_\infty = 1$). Introducing eigenvalues, $\|\tilde{\mathbf{B}}\mathbf{x}_k\|_\infty = \|q_k'\mathbf{x}_i\|_\infty = |q_k'|\ \|\mathbf{x}_i\|_\infty = |q_k'|$ where q_k's are any eigenvalues. Combined with the spectral radius definition, $\breve{\rho}(\tilde{\mathbf{B}}) = \text{Max}_k |q_k'| = \text{max}_k \|\tilde{\mathbf{B}}\mathbf{x}_k\|_\infty$. From the result of Equation A1.50, $\breve{\rho}(\tilde{\mathbf{B}}) \leq \|\tilde{\mathbf{B}}\|_\infty$ for any eigenvalue. This means we have an easily calculable bound of the spectral radius. Thus in the example above, we have $\breve{\rho}(\tilde{\mathbf{B}}) \leq \|\tilde{\mathbf{B}}\|_\infty = 1 + |c|$. This illustrates an important feature of using norms.

According to the definition of the norm of a vector, if $\|\mathbf{x}\|_\infty = 1$, this means that $\text{Max}_i |z_i| = 1$. In this case, the ∞-norm of a matrix is defined as $\|\tilde{\mathbf{B}}\mathbf{x}\|_\infty = \text{Max}_i |\Sigma_j \tilde{b}_{ij} x_j| \leq \text{Max}_i \Sigma_j |\tilde{b}_{ij}||x_j| \leq \text{Max}_i \Sigma_j |\tilde{b}_{ij}|$ (Noble 1969). Hence

$$\|\tilde{\mathbf{B}}\| = \underset{\|\mathbf{x}\| = 1}{\text{Max}} \|\tilde{\mathbf{B}}\mathbf{x}\|_\infty \leq \underset{i}{\text{Max}} \sum_{j=1}^{n} |\tilde{b}_{ij}| \qquad \text{(A1.51)}$$

Suppose the maximum sum occurs at row k^*, then we construct a vector $\mathbf{x}$ with $x_j = 1$ if $\tilde{b}_{k^*j} \geq 0$, and $x_j = -1$ if $\tilde{b}_{k^*j} < 0$. For this $\mathbf{x}$, equality is obtained in Equation A1.51.

Next, define $\mathbf{P}' = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ consisting of linearly independent eigenvectors. We also define the eigenvalue matrix

$$\mathbf{Q}'' = \begin{bmatrix} q_1' & 0 & . & . & . & 0 \\ 0 & q_2' & 0 & . & . & . \\ 0 & 0 & q_3' & . & . & . \\ . & . & . & . & . & 0 \\ 0 & . & . & . & 0 & q_n' \end{bmatrix}.$$

$\tilde{\mathbf{B}}\mathbf{x}_k = \mathbf{Q}''\mathbf{x}_k$ ($\mathbf{x}_k \neq 0$). We have $\tilde{\mathbf{B}}\mathbf{P}' = \tilde{\mathbf{B}}(\mathbf{x}_1, \ldots, \mathbf{x}_n) = (\tilde{\mathbf{B}}\mathbf{x}_1, \ldots, \tilde{\mathbf{B}}\mathbf{x}_n) = (q_1'\mathbf{x}_1, \ldots, q_n'\mathbf{x}_n) = \mathbf{P}'\mathbf{Q}''$. It follows that $\tilde{\mathbf{B}} = \tilde{\mathbf{B}}\mathbf{P}'\mathbf{P}'^{-1} = \mathbf{P}'\mathbf{Q}''\mathbf{P}'^{-1}$. Correspondingly,

$$\tilde{\mathbf{B}}^2 = (\mathbf{P}'\mathbf{Q}''\mathbf{P}'^{-1})(\mathbf{P}'\mathbf{Q}''\mathbf{P}'^{-1}) = \mathbf{P}'\mathbf{Q}''^2\mathbf{P}'^{-1}, \ldots,$$

$$\tilde{\mathbf{B}}^r = \mathbf{P}'\mathbf{Q}''^r\mathbf{P}'^{-1} = \mathbf{P}' \begin{bmatrix} q_1'^r & 0 & 0 & \ldots & 0 \\ 0 & q_2'^r & .. & & \\ & & \ldots & & \\ 0 & \ldots & 0 & q_n'^r & \end{bmatrix} \mathbf{P}^{-1}.$$

Obviously, if $\breve{\rho}(\tilde{\mathbf{B}}) < 1$, in other words, $|q_k'| < 1$ for all k, $q_k'^r \to 0$ as $r \to \infty$. One can conclude therefore that $\lim_{r\to\infty} \tilde{\mathbf{B}}' = \mathbf{0}$ if $\breve{\rho}(\tilde{\mathbf{B}}) < 1$.

For matrix series $\mathbf{I} + \tilde{\mathbf{B}} + \tilde{\mathbf{B}}^2 + \cdots + \tilde{\mathbf{B}}^{k,}$ we have $(\mathbf{I} - \tilde{\mathbf{B}})(\mathbf{I} + \tilde{\mathbf{B}} + \tilde{\mathbf{B}}^2 + \cdots + \tilde{\mathbf{B}}^k) = \mathbf{I} - \mathbf{B}^{k+1}$. From what have been shown, $\breve{\rho}(\tilde{\mathbf{B}}) < \|\tilde{\mathbf{B}}\|_\infty = \text{Max}_i \Sigma_j \tilde{b}_{ij} < 1$ for $\tilde{b}_{ij} \geq 0$.

It follows that $(\mathbf{I} - \tilde{\mathbf{B}})(\mathbf{I} + \tilde{\mathbf{B}} + \tilde{\mathbf{B}}^2 + \cdots + \tilde{\mathbf{B}}^k) = \mathbf{I}$, as $k \rightarrow \infty$. Notice $(\mathbf{I} - \tilde{\mathbf{B}})$ is non-singular, inasmuch as there exists another matrix, namely the matrix series $(\mathrm{I} + \tilde{\mathbf{B}} + \cdots + \tilde{\mathbf{B}}^k)$ such that their product is **I** (non-zero). Hence, we can formally express the series as a finite quantity: $\mathbf{I} + \tilde{\mathbf{B}} + \tilde{\mathbf{B}}^2 + \cdots + \tilde{\mathbf{B}}^k = \mathbf{I}/(\mathbf{I} - \tilde{\mathbf{B}})$ as $k \rightarrow \infty$. The division $\mathbf{I}/(\mathbf{I} - \tilde{\mathbf{B}})$ provides an asymptotic stationary solution to Equation A1.49. It should also be emphasized that the solution is no longer stationary if the spectral radius is equal or bigger than unity. The watershed value of 1 for the spectral radius is referred to as a bifurcation point—a key concept in analyzing system stability. Application of such quasi-deterministic analysis is found in the "Garin-Lowry model" sections in the "Chaos, Bifurcation" chapter of Chan (2005) and also coded in the CD/DVD software under the YiChan directory.

VI. SYSTEM STABILITY

We have demonstrated the concept of bifurcation in the above sections, in which critical values of a parameter determine totally different behavior of the system. We will generalize these concepts in the current section, where system stability is discussed (dePalma and Lefèvre 1987). First we examine the autonomous case of Equation A1.33, in other words, the situation when the system $d\mathbf{X}/dt = \mathbf{F}(\mathbf{X}, \mathbf{H}', t)$ is independent of time, or $d\mathbf{X}/dt = \mathbf{F}(\mathbf{X}, \mathbf{H}')$. With this simplification, we will rewrite Equation A1.33 as

$$\frac{dX_j(t)}{dt} = F_j[X_1(t), \ldots, X_n(t)] \qquad j = 1, \ldots, n \tag{A1.52}$$

In general, it is not possible to solve this system explicitly. Nevertheless, mathematical methods do exist to obtain some information on the solution. Among these methods, we mentioned bifurcation theory, catastrophe theory, and stability theory. The stationary states of Equation A1.52, denoted by $\mathbf{X}'' = (X_1'', \ldots, X_n'')^T$, are defined by $d[X_j(t)]/dt = 0$ ($j = 1, \ldots, n$). They are solutions of the following algebraic system:

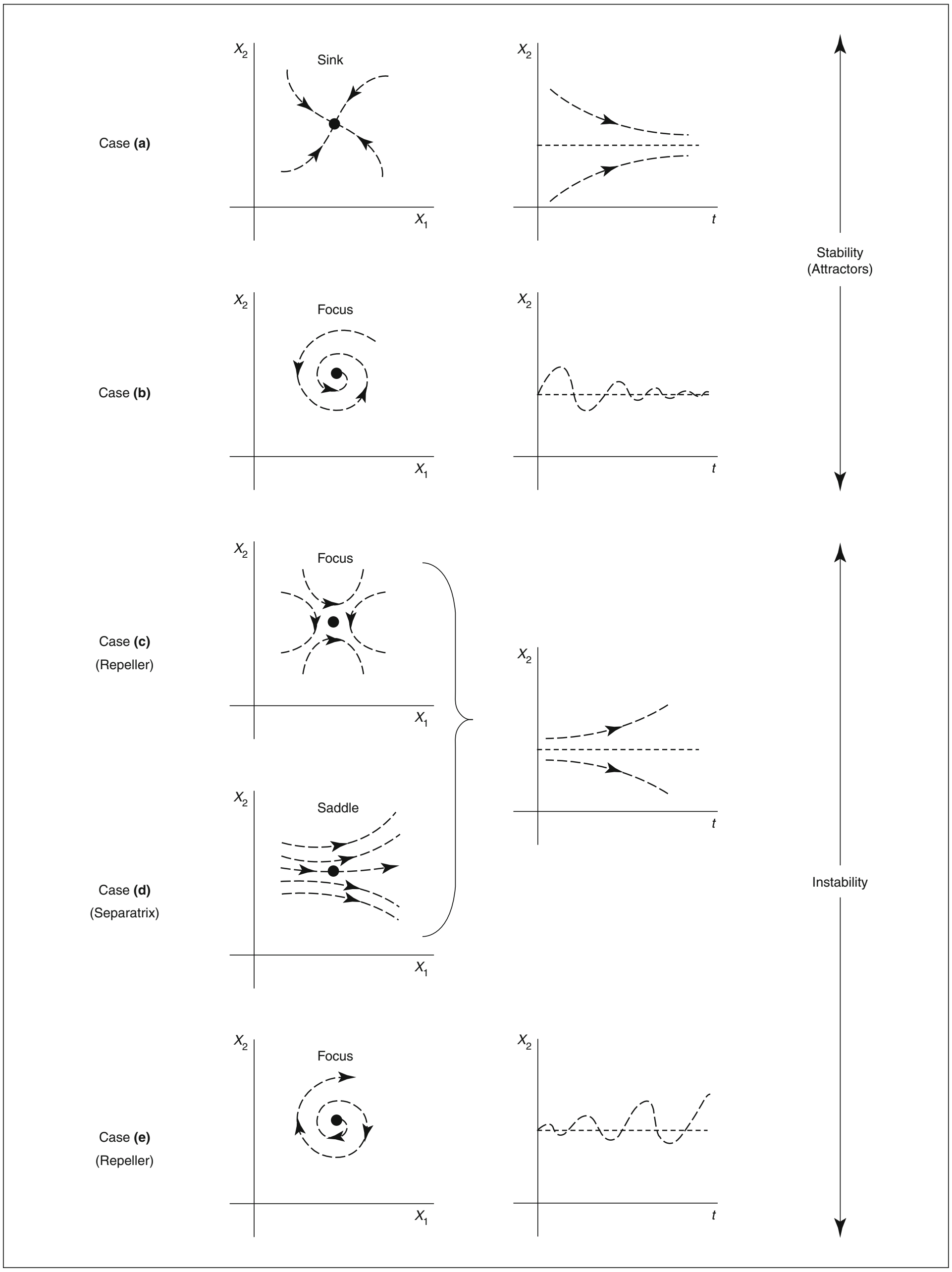
$$F_j(X_1'', \ldots, X_n'') = 0 \qquad j = 1, \ldots, n \tag{A1.53}$$

In other words, these are solutions to the system when motion ceases.

A. Basic Types of Trajectory

The solution as sketched out in Equation A1.53 can be classified into a handful of trajectory types (Wilson 1981). It is also most common for a system to have a small number of single equilibrium points. If they are stable, trajectories lead into them, and they are called **attractors**. If they are unstable, trajectories are repelled by such points, and they are called **repellers**. When there are two or more state variables as shown in Equation A1.53 and sketched out in Figure A1.12, the equilibrium point may be **saddle points,** which represent a special kind of instability. In this case, most trajectories are repelled by such points, but there can be two trajectories (in opposite directions) that pass through the saddle, and these play an

Figure A1.12 STABLE AND UNSTABLE EQUILIBRIA



important role in sketching trajectories in general. They separate the state space into two regions with trajectories on each side being directed to different stable equilibrium points. For this reason, such a trajectory is known as a **separatrix**, and it plays an important role in bifurcation behavior.

To make it perfectly clear, consider a system described by state variables X_1 and X_2. First, we distinguish two kinds of behavior in the neighborhood of a stable equilibrium point. These are shown on a state space plot in Figure A1.12. In case (a), the trajectories lead directly to the equilibrium point and represent an exponential convergence. Such attractors are also called a sink. In case (b), the trajectories spiral into it and represent oscillatory convergence. Such an attractor is also called a focus. Typical time plots for X-against-t are shown also for the two cases side by side.

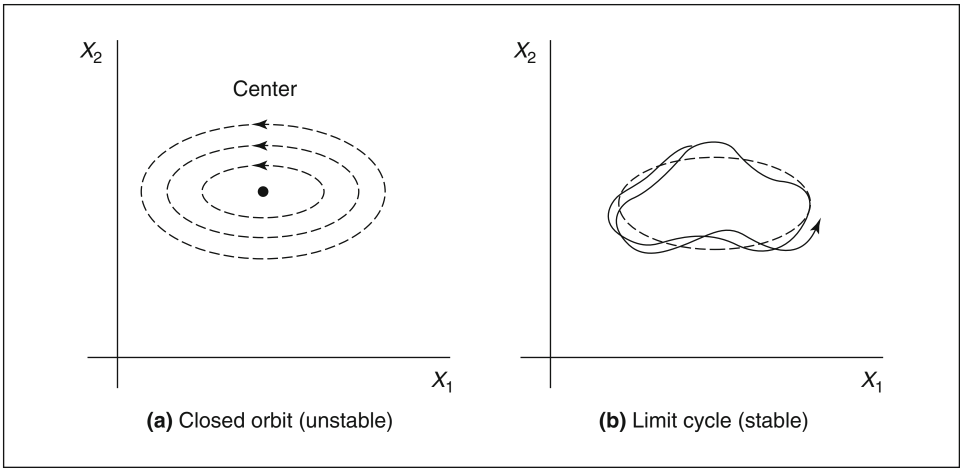
Corresponding plots exist for unstable equilibrium points. Here saddle points behave like unstable case (c) points, but with the addition of the trajectories that form the separatrix as shown in (d). If behavior is neither convergent to a stable point nor divergent, then it may be periodic. There are two basic types as shown in Figure A1.13. Case (a) is a closed orbit periodicity, when the trajectory never leaves one of many possible such orbits (the particular one being determined by the initial conditions). Case (b) is limit cycle behavior: a typical trajectory winds in and out of a closed orbit and may become asymptotically close to it. It turns out that closed orbit behavior is structurally unstable while limit cycle behavior is structurally stable.

Finally, there are examples of system behavior characterized by neither stable or unstable equilibrium points, nor by oscillating behavior of any regular periodicity. Such behavior is called **chaotic** and is demonstrated by irregular looking time plots of state variables. Furthermore, particular (complicated) systems may exhibit a number of different kinds of solution for different starting values of the variables and for different parameter values. As a result, a state-space diagram may be a mixture of trajectories related to different kinds of equilibrium values and may change character as the parameters change.

B. Bifurcation Theory

Bifurcation theory studies the multiplicity of the solutions of Equation A1.53 as a function of some parameters H' of the model (Dendrinos and Mullully 1985; Hildebrand 1962). A bifurcation point $[H', \mathbf{X}''(H')]$ is a point such that in its neighborhood, the multiplicity of the stationary state changes, as discussed above in conjunction with the trajectories and illustrated by the Section V-G

Figure A1.13 PERIODIC TRAJECTORIES IN STATE SPACE



(a) Closed orbit (unstable) **(b)** Limit cycle (stable)

SOURCE: Wilson(1981). Reprinted with permission.

above. Consider a dynamical system of the form $d\mathbf{X}/dt = \mathbf{F}(\mathbf{X}, \mathbf{H}')$ where $\mathbf{X}$ and $\mathbf{F}$ are n-dimensional vectors and $\mathbf{H}'$ is an m-dimensional vector of parameters. As $\mathbf{H}'$ changes, the phase plane[21] also changes. Usually the change is continuous, but at certain bifurcation points the change in dynamic trajectories is abrupt.

The simplest bifurcation is found in the univariate system equation $dX/dt = aX$ as a varies from $-\infty$ to $+\infty$. Negative a generates a set of negative exponential (stable) trajectories, while positive a depicts exponential (unstable) growth. At zero the trajectory bifurcates. These three trajectories and the associated (simple) phase diagram for a is shown in Figure A1.14. In each case, a family of trajectories are shown, corresponding to different initial conditions. We have witnessed an example of this type of bifurcation in Chapter 1, under Figure 1.1.

A slightly more complicated example is the following two-state system

$$\begin{aligned} \dot{F}_1 &= \dot{X}_1 = X_2 \\ \dot{F}_2 &= \dot{X}_2 = X_1^2 - X_2 - a \end{aligned} \qquad \text{(A1.54)}$$

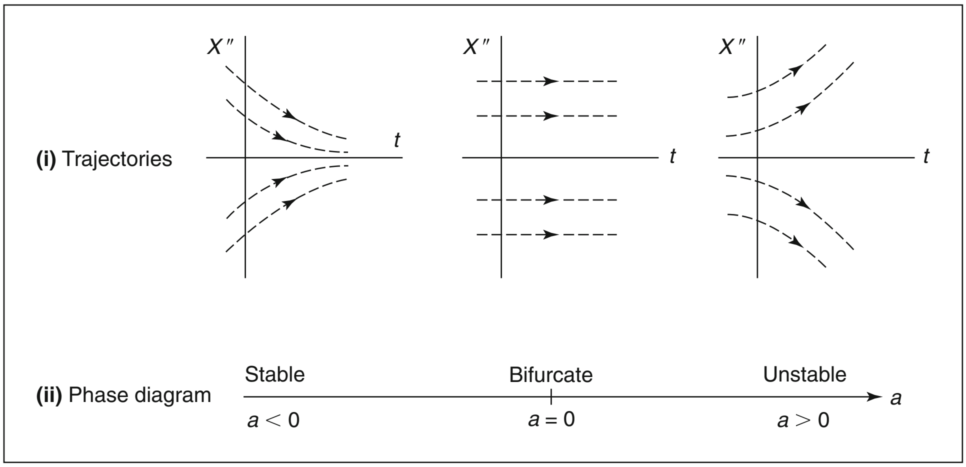
The equilibrium solution is approximated by the matrix linear system $d\mathbf{X}/dt = \mathbf{A}'\mathbf{X}(t)$ where the stability setting Jacobian-matrix $\mathbf{A}'$ has elements A'_{ij} where

$$A'_{ij} = \left.\frac{\partial F_i}{\partial X_j}\right|_{\mathbf{X}''} \qquad i, j = 1, 2, \ldots, n \qquad \text{(A1.55)}$$

In other words, for a two-dimensional case,

$$\mathbf{A}' = \begin{bmatrix} \dfrac{\partial F_1}{\partial X_1} & \dfrac{\partial F_1}{\partial X_2} \\ \dfrac{\partial F_2}{\partial X_1} & \dfrac{\partial F_2}{\partial X_2} \end{bmatrix}_{\mathbf{X}''}$$

Figure A1.14 EXAMPLE OF A SIMPLE BIFURCATION



We follow the solution procedure outlined by Equation A1.35, and the deterministic example worked out in the same Section (V-D). Letting $X_i(t) = X_i'' + \epsilon_i(t)$ and expanding the Taylor series around the equilibrium state X_i'', the solution to the above system is $\mathbf{X}(t) = \mathbf{N}_0 \exp(\mathbf{q}'t)$, where the matrix $\mathbf{N}_0$ contains constants that depend on the initial values $\mathbf{X}(0)$, and $\mathbf{q}'$ is the vector of the eigenvalues of $\mathbf{A}$.

Here in this example, solution to the equations yield $\mathbf{X}'' = (X_1'', X_2'') = (\pm\sqrt{a}, 0)$. If a is negative, there are no real valued equilibria, since the square root of a negative number yields an imaginary root. If a is positive, there are two real valued equilibria: $(\sqrt{a}, 0)$ and $(-\sqrt{a}, 0)$. Linearizing around these points, one obtains the matrix

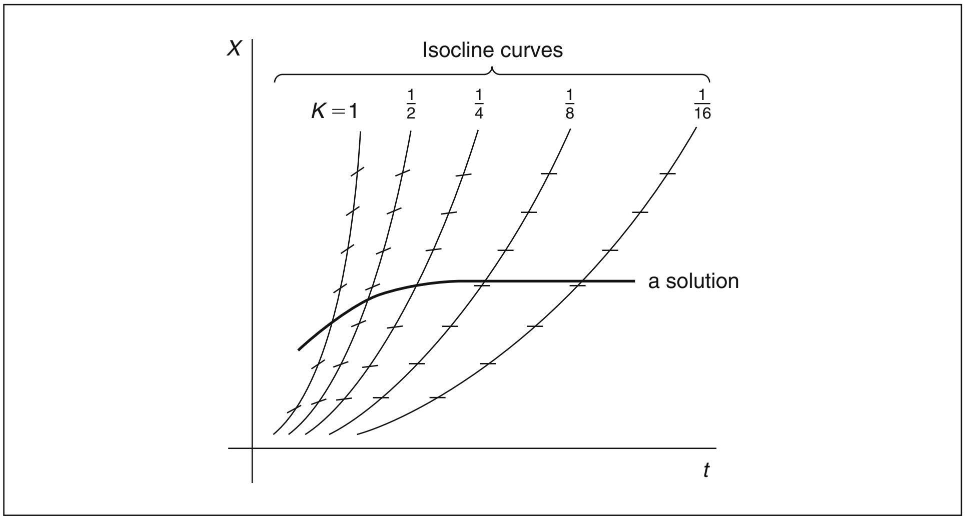
$$\mathbf{A}' = \begin{bmatrix} 0 & 1 \\ \pm 2\sqrt{a} & -1 \end{bmatrix}$$

according to Equation A1.55 with eigenvalues $q_1' = 1/2[-1 \pm (1 + 8\sqrt{a})^{1/2}]$ and $q_2' = 1/2[-1 \pm (1 + 8\sqrt{a})^{1/2}]$. Solution of such system of equations in general yields eigenvalues that are complex or real numbers. The complex part induces an oscillating behavior while the real part gives rise to an exponentially increasing or decreasing solution according to its sign being positive or negative. Consequently, the stationary state $\mathbf{X}''$ is asymptotically stable if all the real parts are negative; the state $\mathbf{X}''$ is unstable if there exists at least one positive real part. In addition, the state $\mathbf{X}''$ is marginally stable if there is at least one eigenvalue whose real part is null and if all the other eigenvalues have a negative real part.

Consider the ordinary differential equation $dX/dt = F(X, t)$. Remembering isoclines are the family of curves defined by the equation $F(X, t) = K$, where K is a constant. In the autonomous case under consideration, this becomes simply $dX(t)/dt = F(X(t))$ and $F(X(t)) = K$. The differential equation states that at any point $X(t)$ for which $F(X(t))$ is defined, the slope of any integral curve passing through that point is given by $F(X(t))$. Suppose we plot the family of isocline curves, $F(X(t)) = K$ for a series of values of the constant K. All integral curves of the differential equation intersect a particular curve of the family of isocline curves with the same slope angle α, where tan α is given by the value of K specifying the isocline. Thus if on each isocline a series of short parallel segments having the required slope is drawn, an infinite number of integral curve can be drawn by starting in each case at a given point on one isocline and sketching a curve passing through that point with the indicated slope and crossing successive isoclines with the slopes associated with them. This method can always be used to determine graphically the particular solution of the differential equation that passes through a prescribed point $X^*(t)$ when the function F is single valued and continuous. The procedure is illustrated in Figure A1.15.

Applying the above procedure to the current two-state example, the first intersection of isoclines always implies a saddle, since $1 + 8\sqrt{a} > 0$. One eigenvalue is positive whereas the other is negative. However, in the second intersection $1 - 8\sqrt{a}$ could be positive, zero, or negative. At the point where it is zero, ($a = 1/64$), the nature of the dynamic path changes. As a increases the system's trajectories are transformed from a stable sink (q_1', q_2' negative, real, unequal) to a stable focus (q_1', q_2' complex, with negative real parts). Another illustration is found under the "Synergetic Models of Spatial Interaction" subsection of the "Activity Allocation and Derivation" chapter in Chan (2005).

Figure A1.15 GRAPHICAL SOLUTION OF A DIFFERENTIAL EQUATION



C. Comments

The above example is cited for illustrative purposes only. In real systems, problems tend to be nonlinear, and the solution procedure becomes a lot more complex. It is not unusual to resort to numerical simulation, which is often the only feasible means to solve the problem. Insights can be obtained, however, by developing a qualitative theory in which the topological nature of the model's equilibrium point is studied. In the preceding subsections, we have explored how the main types of solutions for dynamical systems described by differential equations are constructed and how to get some insights by representing them graphically or through simple examples. It should already be clear by implication that the possible types of bifurcation behavior are richer than indicated in the canonical forms of catastrophe theory. We now summarize several cogent observations (Wilson 1981).

First, we note that the solutions (for equilibrium points) to Equations A1.52 will typically involve multiple solutions because of any nonlinearities in the functions F_j. Hence, the manifold of equilibrium solutions in the space of ($\mathbf{X}$, $\mathbf{H}'$) variables will be folded. This can lead to the same broad kinds of bifurcation as in catastrophe theory, as suggested in the introduction to Section IV.

Second, we observe that the types of solutions to the differential equations can be determined by parameter values. There can be critical parameter values at which a stable sink becomes a focus, as shown in the numerical example above. Similarly, one can envisage situations in which a stable equilibrium point becomes unstable or disappears, or at which a periodic solution could disappear and be replaced by a stable equilibrium point, or vice versa. (These changes are collectively known as the **Hopf bifurcation**). In theory, all the possible interchanges between the kinds of solution (or trajectory) listed in Section VI-A are possible. It is useful to be alert to this in applied work.

Finally, we also note here in passing a completely different type of possible bifurcation. Suppose a system is disturbed from an equilibrium position and moves to a non-equilibrium state in the neighborhood of a separatrix in state space. Then if the separatrix is crossed, the return to equilibrium could be to a state other than the original one. This is analogous to the cusp catastrophe discussion in Subsection IV-A.

VII. CONCLUDING REMARKS

In this appendix, we have reviewed the theories that govern the evolution of complex systems over time, including the influence of external factors. It can be seen that the body of knowledge in this area is huge and has diverse roots among a number of disciplines. We can, at best, provide only an overview in the limited number of pages here. While the theories hold great promise in modeling facility location and land use, their present status can often only allow us to computationally solve small problems. In limited cases, the theories can be best used qualitatively to gain insights, rather than used quantitatively to yield computational results, even for small problems. Many larger systems need to be modeled by numerical simulation, even though such systems can be set up analytically as equation sets. This point is demonstrated in some detail in the main body of this book and Chan (2005). Under certain circumstances, stochastic systems can be approximated by deterministic systems, effecting a fair amount of computational savings. Overall, the aim of this appendix is to provide the basics and a road map for readers to relate these diverse theories to one another, particularly in the context of problem solving. We include the appropriate references to the vast amount of literature for further investigation.

ENDNOTES

[1] For the dynamic programming problem, this is called the state transition equation.

[2] Those who are familiar with the calculus of variations of deformable bodies will recognize this as the variational form of a thin membrane, often written as $\boldsymbol{\delta}\int_{\Omega} 1/2\ (\nabla u)^2\ d\mathbf{x} = 0$, where u is the amplitude of such small deformation as oscillation. Here u takes on a prescribed function along the boundary. In this context, the variational equational equation simply prescribes that the potential energy stored in the membrane must be in equilibrium. In the two-dimensional x_1-x_2 case, for example, the potential energy $1/2(\nabla u)^2$ is simply $1/2(u_x^2 + u_y^2)$.

[3] Suppose that a function $f(X(t), U(t))$ exists such that $f_X \equiv F(X, \mathrm{U}) = \partial X/\partial t = \dot{X}$. A dynamic system that can be derived from such a function $f(X, U)$ is formally called a gradient system.

[4] The co-rank measures the degree of degeneracy of the worst kind of singularity that can occur in the particular family of functions. For nth-order polynomial functions of one variable, for example, the degree of degeneracy is n, where all derivatives up to the nth order vanish.

[5] The co-dimension of a family of functions is the number of control variables U that parameterize these functions.

[6] An example of such an application can be found in the "Chaos, Catasrophe, Bifurcation and Disaggregation" chapter in Chan (2005) under the "Spatial Dynamics" section.

[7] For a discussion of the Markovian process, see Appendix 3.

[8] When the column vectors in a square matrix are mutually orthogonal and of unit length, we say that the matrix is orthonormal. Specifically, if the dot product of vector $\mathbf{x}$, $\mathbf{x}^T\mathbf{x} = \|\mathbf{x}\|^2 = 1$, the vector $\mathbf{x}$ is said to be normalized. If a set of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ is orthogonal and normalized (in other words, $\mathbf{x}_i^T\mathbf{x}_j = 0$, $i \neq j$; $\mathbf{x}_i^T\mathbf{x}_i = 1$), then the vectors are said to be orthonormal.

[9] An example of such equations has been illustrated in Appendix 3 in the derivation of the Poisson process.

[10] The logit model is explained in Chapter 3.

[11] An example of the birth-death process has been illustrated in the derivation of the *M*/*M*/1 queue as part of Appendix 3.

[12] Consider a random variable taking on the values $\xi = 0, 1, 2, \ldots$ with the associated probabilities P_0, P_1, P_2, ... The generating function for this probability distribution is $G(\xi) = P_0 + \xi P_1 + \xi^2 P_2 + \cdots = \sum_x \xi^x P_x$. This function has several useful properties. First $G(1) = 1$, $G(0) = P_0$, and $dP/d\xi = \sum_x \boldsymbol{x}\xi^{x-1} P_x$. Notice that $dG/d\xi|_{\xi=1} = \sum_{x=0}^{\infty} x\, P_x = \mathrm{E}(x)$, which yields the mean of the random variables *x*. By the same token, $d^kG/d\xi^k|_{\xi=0} = x!P_x$ which yields the individual terms of the distribution. The generating function is often used to derive many analytical results in stochastic processes.

[13] Both the predator-prey and the dynamic Lowry derivative models are explained in the "Lowry-based Models" and "Chaos, Catastrophe, Bifurcation and Disaggregation" chapters of Chan (2005).

[14] A homogeneous equation is one that does not have an input or forcing function $U(t)$ on the right-hand side. In the case of linear algebraic equations, this means the right-hand side is a zero vector. For a linear homogeneous equation, any linear combination of individual solutions is also a solution.

[15] A particular solution to a differential equation is that part of the solution in response to the input or control function $\mathbf{U}(t)$.

[16] Consider the homogeneous set of equations: $\mathbf{AX} = q'\mathbf{X}$. Values of q' for which non-trivial solutions exist are called eigenvalues, and corresponding vector solutions $\mathbf{X}$ are known as eigenvectors. More specifically, $\mathbf{X}$ here is a right eigenvector.

[17] If $(\mathbf{A}' - q'\mathbf{I})\mathbf{x} = \mathbf{0}$ where $\mathbf{x} \neq \mathbf{0}$, and we set the determinant to zero, in other words, $|\mathbf{A}' - q'\mathbf{I}| = 0$, then the scalar roots q'_k of the resulting polynomial are eigenvalues of $\mathbf{A}'$.

[18] If the row and column containing an element (i, j) in a square matrix are deleted, the determinant of the remaining square array is called the minor of (i, j), and is denoted by M_{ij}. The *cofactor* of (i, j) is then defined by $(-1)^{i+j}M_{ij}$.

[19] This can be likened to a time series that consists of a structural part and a noise term. See the "Spatial Time Series" chapter of Chan (2005) for a more complete discussion.

[20] This is witnessed by the solution to the Garin-Lowry model as shown in the "Chaos, Catastrophe, Bifurcation, and Disaggregation" chapter of Chan (2005). In the un-capacitated case, a constant transition matrix Π is assumed, resulting in the special solution of Equation A1.46.

[21] Also known as a phase diagram, this is an analytic device to characterize the solution without necessarily writing out the system solution explicitly. An example will be illustrated shortly in Figure A1.14.

REFERENCES

Belensky, A. S. (1998). *Operations research in transportation systems: Ideas and schemes of optimization methods for strategic planning and operations management*. Boston: Kluwer Academic Publishers.

Chan, Y. (2005). *Location, transport and land-use: Modeling spatial-temporal information.* Berlin and New York: Springer.

Cox, D. R.; Miller, H. D. (1965). *The theory of stochastic processes.* New York: Wiley.

De Palma, A.; Lefèvre, C. L. (1987). "The theory of deterministic and stochastic compartmental models and its applications." In *Urban systems: Contemporary approaches to modelling,* edited by C. S. Bertuglia, et al. London: Croon Helm.

Dendrinois, D. S.; Mullally, H. (1985). *Urban evolution: Studies in the mathematical ecology of cities.* Oxford and New York: Oxford University Press.

Godfrey, K. (1983). *Compartmental models and their application.* New York: Academic Press.

Haag, G. (1989). *Dynamic decision theory: Applications to urban and regional topics.* Boston: Kluwer Academic Publishers.

Hildebrand, F. B. (1965). *Advanced calculus for applications.* Englewood Cliffs, New Jersey: Prentice-Hall.

Kinderlehrer, D.; Stampacchia, GT. (1980). *An introduction to variational inequalities and their applications.* New York: Academic Press.

Lorenz, H. W. (1993). *Nonlinear dynamical economics and chaotic motion,* 2nd ed. New York: Springer-Verlag.

Minoux, M. (1986). *Mathematical programming-theory and algorithms.* (Translated by Stephen Vajda). New York: Wiley.

Nagurney, A. (1993). *Network economics: A variational inequality approach.* Boston: Kluwer Academic Press.

Noble, B. (1969). *Applied linear algebra.* Englewood Cliffs, New Jersey: Prentice-Hall.

Seber, G. A. F.; Wild, C. J. (1989). *Nonlinear regression.* New York: Wiley.

Silberberg, E. (1990). *The structure of economics: A mathematical analysis,* 2nd ed. New York: McGraw-Hill.

Wilson, A. G. (1981). *Catastrophe theory and bifurcation: Applications to urban and regional systems*. London: Croom Helm and Berkeley, California: University of California University Press.

Yi, P. (1986). Infrastructure management: A bifurcation model in urban regional planning. Master's Thesis. Department of Civil and Environmental Engineering. Washington State University. Pullman, Washington.

Appendix 2

Review of Some Pertinent Statistical Tools

This appendix puts in one place a few basic statistical analysis techniques, including estimators, goodness-of-fit parameters, ordinary and stepwise regression, analysis of variance, nonlinear regression, and the general idea behind statistical modeling. As with the previous two appendices, we strive to provide a self-contained account through numerical examples, rather than formal developments. Most important, the relationships between different statistical tools are clearly delineated, particularly in our explanation of stepwise regression. It paves the way for chapters such as Chapter Three and these chapters in Chan (2005): "Generation, Competition and Distribution," "Spatial Econometrics," "Spatial Time Series," and "Spatial Temporal Information." It is particularly convenient in numerous places in the current book where statistical knowledge is assumed, including the software on the CD/DVD.

I. STATISTICAL ANALYSIS: BASIC CONCEPTS

For the purpose of this discussion, statistics can be thought of as dealing with representative indicators of figures, when a huge number of figures need to be summarized in terms of a more compact set of information. An estimator such as the mean or average is a good example, wherein n numbers are represented in terms of a single one: $\bar{X} = \sum_{i=1}^{n} xi/n$. Here capital X stands for the random variable for the data and x_is are the data observations themselves. Similarly, one can define the spread of the data about the mean, an estimator called standard deviation:

$$s = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{X})^2}{(n-1)}} \tag{A2.1}$$

Variance is the square of standard deviation, such that the sign of standard deviation—indicating whether the specific figures are smaller than or bigger than the mean—is set aside. To summarize the two estimators, mean and standard deviation, one can define the **coefficient of variation**, which compares the magnitude of the *spread* with the *average*, $s/\bar{X}$. A small coefficient suggests a sharp distribution, while a large one implies a flat distribution.

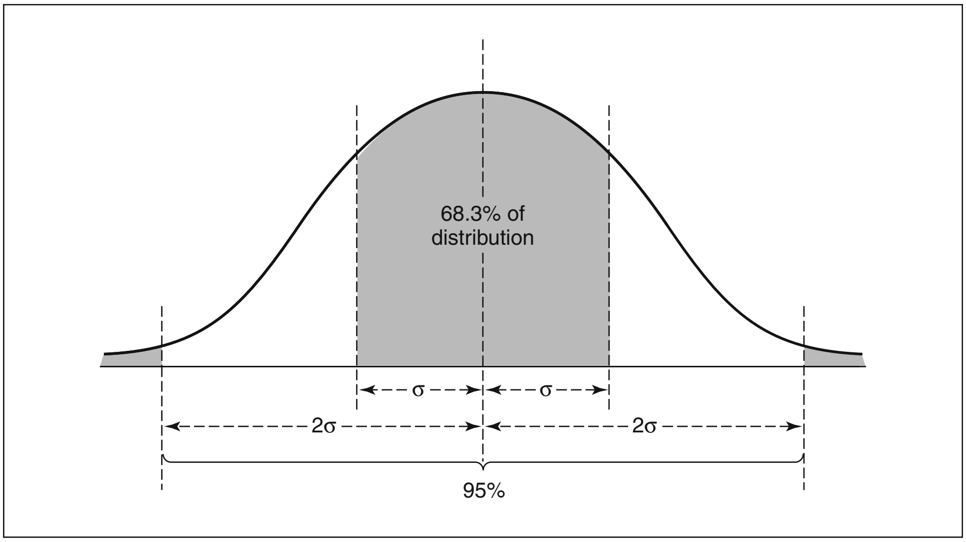
A normal distribution is representative of many large samples of data—data on anything from income to population. The mean and standard deviations are overlaid on top of the normal distribution in Figure A2.1. It shows that about 68.3 percent of the sample will be within one standard deviation from the mean and about 95 percent within two standard deviations. Inasmuch as the normal distribution describes any large sample, such numbers are very useful in detecting abnormalities such as outliers.

It is appropriate at this time to introduce the concept of **degree of freedom** (*dof*). Notice that in computing the standard deviation, we divide the sum of the data by $(n - 1)$ instead of n, which makes it different from computing the mean. While there is a lengthy explanation possible for such a practice, there is an informal way to rationalize it here. We can think of the degree of freedom as the number of useful, or contributory, pieces of information. Imagine that a piece of information is no longer useful once it has been used. If there are n data points to begin with, or n useful pieces of information. We extract from the pool of data one piece of information, for instance, say the mean. The number of unused ones or contributory data remaining will be $(n - 1)$. Viewed in this light,

$$dof = (\textit{number of observations} - \textit{number of coefficients estimated})$$

It goes without saying that the larger the degree of freedom, the more representative the estimator, since it is based on a large pool of useful information, instead of a meager sample. In practice, there are often more data pieces than the number of estimated parameters; the precise *dof* becomes less important and the difference between $(n - 1)$ and n in the denominator of Equation A2.1 is minute. We suggest that the sample estimator [using $(n - 1)$ division] approximates the population estimator [using n division].

Figure A2.1 ILLUSTRATING MEAN AND STANDARD DEVIATION IN A NORMAL DISTRIBUTION



II. GOODNESS-OF-FIT MEASURES

If one generalizes from a single dimension to two or more dimensions, we start to worry about the relationship between the data represented in these dimensions. For example, we may be interested in the linkage between the number of work trips made and the population density in each of the subareas. Likewise, we may want to study the relationship between the employment level and the population, suspecting that the two may be related. In other words, the more people around, the larger the number of people working and hence the more work trips made. We call these the degrees of correlation. The more one variable is related to another, the larger the correlation between them. The **correlation coefficient**, *r*, between the retail-land-use random variable (Y) and the retail-employment random variable (X), for example, may be 0.96. This indicates a close relationship between the two variables, considering that by convention, 1 is the largest correlation possible;

$$r_0 = \frac{\Sigma_t\,(x_t - \overline{X})\,(y_t - \overline{Y})}{s_X s_Y}$$

where x and y are data observations. A close examination of this expression will suggest that a plot of the retail land use (in acres or hectares) against retail employment will yield a linear relationship, in that retail land use goes up as retail employment goes up. One can readily see this in a shopping mall, as illustrated in the correlation coefficient plot in Chapter 3.

In forecasting applications, we often distinguish between two situations. The first is when the relationship between two variables are sought, both of which are known for a future year. The second is when we want to forecast a variable from an independent variable that we know. The relationship between the latter pair of variables is called the **partial correlation coefficient** while the relationship between the former is simply the correlation coefficient. Another way of saying this is that the partial correlation coefficient measures the linear association between the **dependent** and **independent** variables, while the correlation coefficient does the same job between two independent variables. We want a high value for the partial correlation coefficient. The exact opposite is true for the correlation coefficient. The reason is that a high partial means good explanatory power of the independent variable in predicting the dependent variable. However, a high correlation among two independent variables means that there is some kind of double counting. In other words, the same information is used twice to predict the dependent variable. For example, one should be careful in using both population and employment as independent, or explanatory, variables for the number of work trips generated from a subarea. The simple reason is that they are related. Including both variables will lead toward a statistical fallacy known as **collinearity**.

If we generalize the concept to two or more independent variables and one dependent variable, we have broadened the concept of a partial correlation coefficient to a **multiple correlation coefficient**. This pertains to the power of several independent variables in predicting the dependent variable. For example, if land use development is to be forecasted, one way to do this is to relate land development in the future to the population and per capita income in the area.

The stronger the relationship between land development and population/income, the more accurate the forecast is likely to be. In other words, a multiple correlation coefficient R close to unity is preferred to one that is smaller. When there is only one dependent and one independent variable, the multiple correlation coefficient R becomes the partial correlation coefficient r'_0.

III. LINEAR REGRESSION

The concept of multiple correlation coefficient, the square of which is sometimes known as **coefficient of multiple-determination**, brings us to the subject of **linear regression**. Linear regression can be thought of as the structural postulation between dependent and independent variables that can be supported by sound **goodness-of-fit** parameters, where goodness-of-fit parameters are simply multi-dimensional extension of estimators like mean and variance. Take the example of work trip forecasting. Suppose the following structural relationship is postulated: $Y = a + bX$ where Y is the number of total trips predicted and X is the household income (in thousands of U. S. dollars), and a, b are calibration coefficients, which take on the values of 29.33 and 1.150 respectively for the data shown in Table A2.1. These values are calculated on the basis of these formulas:

$$b = \frac{\sum_{t=1}^{n} (x_t - \overline{X})(y_t - \overline{Y})}{\sum_{t=1}^{n} (x_t - \overline{X})^2}$$

and $a = \overline{Y} - b\overline{X}$. In other words, b is calculated as

$$\frac{(30 - 20)(65 - 52.333) + (20 - 20)(50 - 52.333) + (10 - 20)(42 - 52.333)}{(30 - 20)^2 + (20 - 20)^2 + (10 - 20)^2}$$

and a as 52.333 − (1.15)(20). These formulas determine a and b on the basis of minimizing the sum of the deviations of the dependent variable from the regression line.

To show this concept, a plot of the regression line is given in Figure A2.2, in which the deviations, sometimes referred to as residuals, are highlighted. Here the multiple correlation coefficient, R, is 0.985. This is close enough to 1.000. In an applicational context, however, this coefficient is often much less than unity. The toy problem we have been using, as illustrated in Figure A2.2, shows a definite relationship between work trip and population. In most regression applications, the square of the figure is used, R^2, such that

$$R^2 = \frac{\sum_{t=1}^{n} (\hat{y}_t - \overline{Y})^2}{\sum_{t=1}^{n} (y_t - \overline{Y})^2}$$

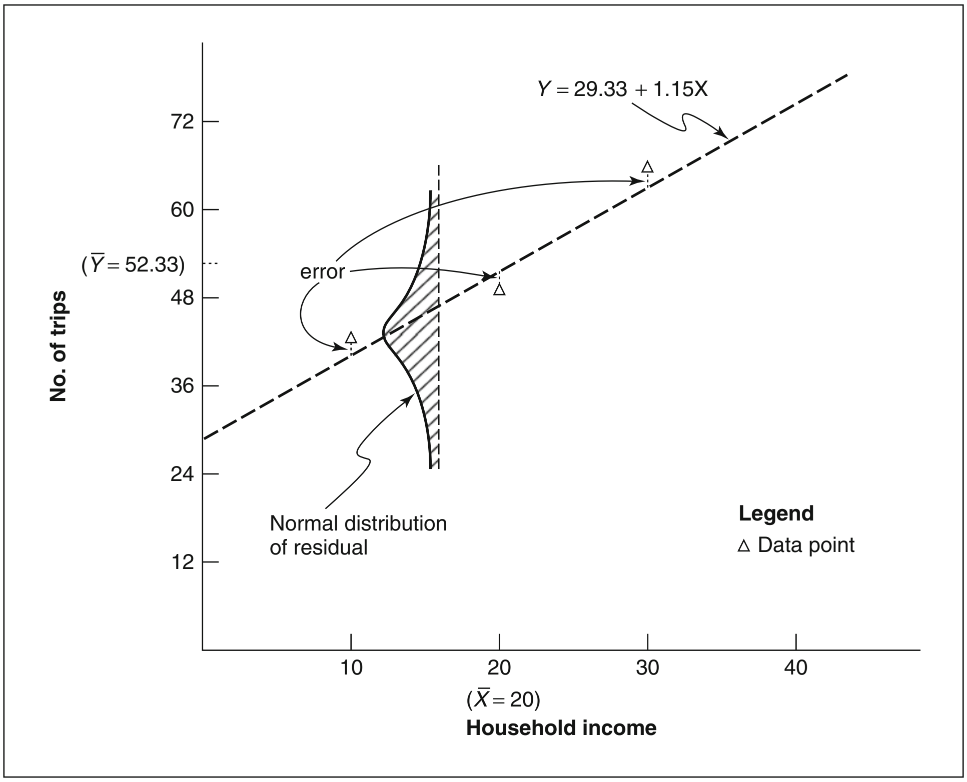
Table A2.1 DATA FOR THE REGRESSION EXAMPLE

Households	Household income X (in thousands)	No. of trips per week Y	Estimated no. of trips $\hat{Y}$	Error ϵ
Jones	30	65	63.833	1.167
Browns	20	50	52.333	−2.333
Robinsons	10	42	40.833	1.167

where $\hat{y}_t$ is the estimated number of trips for family t ($t = 1, 2, 3$) from the equation $\hat{Y} = 29.33 + 1.15X$. In other words, they are the values read off from the regression line itself (Figure A2.2) for a given household income x. $R^2 = 0.970$ is then calculated as

$$R^2 = \frac{(63.833 - 52.333)^2 + (52.333 - 52.333)^2 + (40.833 - 52.333)^2}{(65 - 52.333)^2 + (50 - 52.333)^2 + (42 - 52.333)^2}$$

Figure A2.2 REGRESSION LINE OF EXAMPLE



One can think of $\hat{Y}$ as a two-dimensional generalization of the mean estimator. The difference between the data point y_t and the estimated value $\hat{y}_t$ is called the error of estimation a_t. Viewed in this light, one can write $y_t = \hat{y}_t + a_t$, and the regression equation can be written as $Y = a + bX + \epsilon$ where ϵ is the error term random variable (see Figure A2.2 and Table A2.1).

The parameter R^2 has the interpretation of the percentage of variation explained by the regression. In other words, the amount of relationship that is captured by the linear model itself in terms of the structural equation $Y = a + bX$, with the remaining part due to random error associated with any statistical analysis. A moment's reflection will show that R^2 can also be expressed in terms of these alternative expressions

$$R^2 = \frac{(\textit{source of variation due to regression})}{(\textit{total source of variation})}$$

or

$$R^2 = \frac{(\textit{variance explained by the regression})}{(\textit{total unconditional variance})}$$

The higher the R^2 value, the more significant the regression equation is. The only exception is over-fitting, which is best exemplified by fitting two data points with a regression line. This results in $R^2 = 1$, but $dof = 0$, meaning there is no allowance for statistical analysis. As seen from this toy problem and will be shown in the analysis-of-variance discussion later, including a large number of independent variables (in comparison to the number of data points) decreases the *dof* and increases the R^2. The increase in R^2 purely due to a larger number of independent variables is not necessarily helpful. First, there is an extra cost of data collection. Aside from data collection, there is also an extra burden in using a more complicated model. Second, including two independent variables that are strongly correlated does not contribute to the explanatory power of the equation. As a matter of fact, it will detract from it. There is a delicate tradeoff, therefore, between having a perfect statistical fit and simplifying a model by minimizing the number of explanatory variables. We often refer to this tradeoff as the art of parsimony.

The standard error of estimate (SEE) is a measure of the dispersion of the observed data about the regression line. This can again be thought of as a two-dimensional generalization of the standard deviation about the mean. The smaller the SEE, the tighter the fit of the regression line about the data:

$$SEE = \sqrt{\sum_{t=1}^{n} \frac{(y_t - \hat{y}_t)^2}{dof}}$$

It can be verified that the *SEE* in this case is 2.858, which is calculated as

$$SEE = \sqrt{[(65 - 63.833)^2 + (50 - 52.333)^2 + (42 - 40.833)^2]/(3 - 2)}$$

Two dimensional generalization of the coefficient of variation can also be made. It is simply

$$\left(\frac{SEE}{\overline{Y}}\right) 100\%$$

It measures how accurately the dependent variable can be estimated by the regression equation, relative to the mean of the dependent variable observations. Obviously, the smaller the ratio, the more accurate the estimate tends to be. In our case, the ratio is (2.858)/(52.333) = 5.46%, reflecting quite a high degree of accuracy.

A last set of goodness-of-fit measures pertains to the regression coefficients. The *t*-ratio or *t*-statistic shows how well the coefficient *b* is calibrated. Statistically, it is simply defined as $t_b = b/s_b$, where $s_b = SEE/(s_X\sqrt{n})$ when *n is a large number*. Notice this is again a two-dimensional generalization of the concept of $s_{\overline{X}} = s_X/\sqrt{n}$. This is a test on the *null hypothesis* that the coefficients should be zero. In other words, the regression equation has no explanatory power since the data has no pattern, or the data represent total randomness. In this toy example, t_b is calculated as

$$t_b = \frac{1.150}{2.858/(10)(\sqrt{3})} = 6.969$$

While the first equation in this paragraph may be inappropriate for a small number of data points ($n = 3$) in this case, the formula should be a good approximation for large samples in general. To assess how significant the calibration coefficient *b* is, we examine the *t*-table, which shows that *t* (1, 0.90) = 6.314, or the *t*-value for a Student's *t* distribution at 1 *dof* and 90 percent confidence level is 6.314. Since $t_b = 6.969$ is larger than 6.314, we reject the null hypothesis and state that the coefficient *b* is significant at 90 percent confidence level. In other words, the linear regression model is useful in explaining the variation of the *Y* random variable in terms of the *X* random variable. Implicit in the definition of the *t*-statistic is the assumption that the error is normally distributed. This is illustrated in Figure A2.2 by the normal distribution drawn around the regression line. The technical way of describing this assumption is to say that the residuals are **homoscedastic**.

F-ratio or *F*-statistic measures how well the coefficients perform as a whole, in this case only the parameter *b* itself. *F* is computed as

$$F = \frac{\textit{mean variance due to regression}}{\textit{mean variance about regression}} = \frac{\sum_{t=1}^{n} (\hat{y}_t - \overline{Y})^2/dof}{SEE} \qquad \text{(A2.2)}$$

Here the *dof* refers to the regression coefficient data pool rather than the observation data pool. The *F*-statistic is then calculated as

$$F = \{[(63.833 - 52.333)^2 + (52.333 - 52.333)^2 \\ + (42 - 53.333)^2]/(2 - 1)]\}/(2.858)^2 = 32.388$$

Since there is only one coefficient b in a bivariate regression, the square of the t-ratio is the same as the F-value. To show the significance of the calibration, we examine the F-statistic, $F(1,1,0.90) = 39.9$ (or the F-ratio at 90 percent confidence level with 1 *dof* at the numerator and the denominator. This supports the null hypothesis, suggesting that the calibration parameters are insignificant. However, with 32.388 being bigger than $F(1, 1, 0.75) = 5.83$, it says that the calibration parameters are significant at a reduced confidence level of 75 percent, if in fact 75 percent confidence level is acceptable. Thus by lowering the confidence level, a formerly unacceptable regression model may now be acceptable.

IV. ANALYSIS OF VARIANCE

To gain better insight, an analysis of variance can be performed on linear regression. Analysis of variance (ANOVA) breaks total variance of a set of data into two components: data dispersion from local mean and local mean deviation from global mean. Placed in the context of regression, one can explain the sum of squares of the deviations of y_t about Y'' in terms of the sum-of-squares of the deviations of $y_t s$ from the regression line and the sum-of-squares of deviations of the estimated values $\hat{y}_t$ about $\overline{Y}''$:

$$\textstyle\sum_t (y_t - \overline{Y})^2 = \sum_t (y_t - \hat{Y})^2 + \sum_t (\hat{y}_t - \overline{Y})^2 \qquad \text{(A2.3)}$$

This equation is best illustrated by Figure A2.3, which breaks down total variance into its two components for an illustrative data point (x_t, y_t). Another way to explain ANOVA for regression is that

(total [corrected] sum of squares) = (error sum of squares)
+ (explained sum of squares)

or

(total source of variation) = (source of variation about regression)
+ (source of variation due to regression)

The word corrected is used to distinguish between raw data y_t and data corrected for the mean $(y_t - \overline{Y})$. The degrees of freedom for each of the above three terms are $n - 1$, $n - k$, and $k - 1$ respectively, where k is the number of parameters estimated in the regression equation.

A typical ANOVA table is shown in Table A2.2 for the example problem we have been using thus far. It can be verified that the F-ratio is computed as $264.5/8.167 = 32.388$, which is exactly the same as Equation A2.2. When the numerator (mean variance due to regression) possesses only one *dof*, as in the case of a bivariate regression, F will be the same as t^2 (as mentioned). This is checked out in this case, where the t-statistic was computed as 5.691 ($t^2 = 32.388$).

Figure A2.3 ANALYSIS OF VARIANCE AS APPLIED TO LINEAR REGRESSION

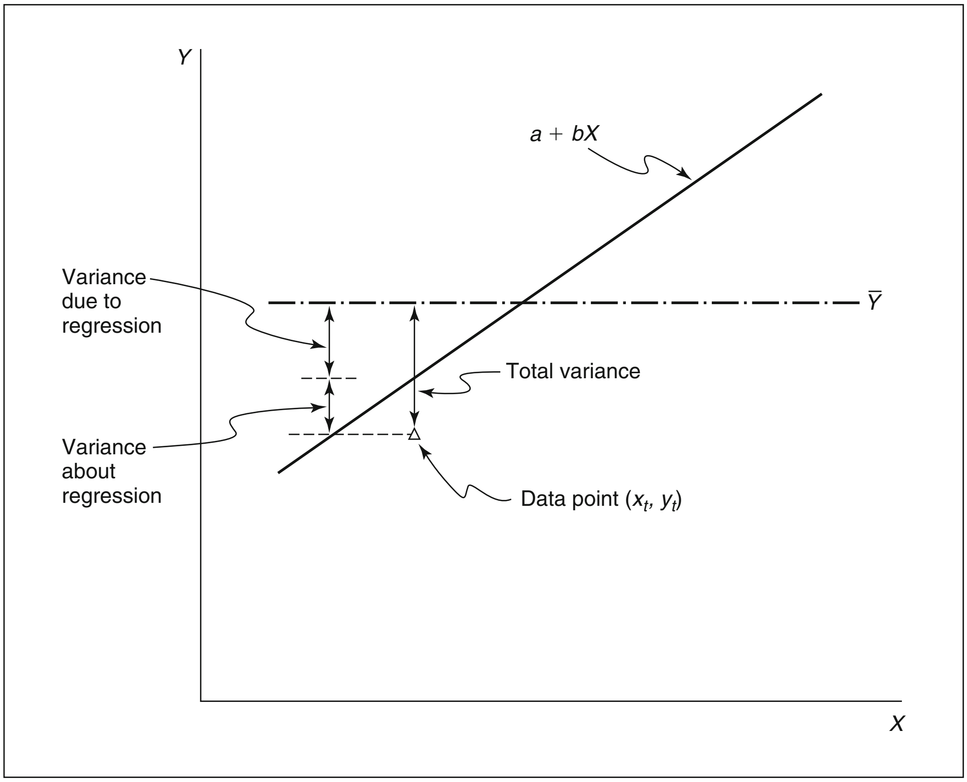

Y
a + bX
Variance due to regression
Ȳ
Total variance
Variance about regression
Data point (x_t, y_t)
X

Table A2.2 EXAMPLE ANALYSIS-OF-VARIANCE TABLE

Source of variation	Degree of freedom	Sum of squares	Mean square	*F*-ratio
Due to regression	1	246.500	264.500	32.388
About regression	1	8.167	8.167	
Total	2	272.667		

V. USING THE REGRESSION EQUATION

Earlier, we made a distinction between sample estimator and the population estimator. An analogy can be drawn for the regression line here. A model can be constructed to estimate the "true" Y values from a population regression line $E[Y \mid X = x^*] = \alpha' + \beta' X$ (Crow, Davis, and Maxfield 1960). This contrasts with an estimate obtainable from a regression line $y = a + bx$ based on sample observations (x^*, y^*). Since a and b are observations from random variables $\tilde{a}$ and $\tilde{b}$, the estimator for y^* is $\hat{y} = a + bx^*$. In the absence of a true population, we are interested in the accuracy of estimating $E[Y]$ at $X = x^*$.

A. Confidence Interval

In practical applications, the ordinate of the sample regression line for any given x^* (which need not be any of the observed x_i's) is calculated as $Y = a + bX$. This Y necessarily differs from the true or population mean ordinate at $X = x^*$, which would be obtained if an infinite number of observations could be made with the same value x^*. We can show how good our estimate Y of the true mean ordinate $E[Y \mid X = x^*]$ is by calculating the $100(1 - \alpha)\%$ confidence limits

$$\begin{aligned}\hat{Y} \pm t_{\alpha/2,\,n-2}\,\sigma_{M^*} &\simeq \hat{Y} \pm t_{\alpha/2,\,n-2}\,s_Y \sqrt{\frac{1(x^* - \overline{X})^2}{n + \Sigma_t(x_t - \overline{X})^2}} \\ &= \hat{Y} \pm t_{\alpha/2,\,n-2}\,s_Y \sqrt{\frac{1}{n} + \frac{(x^* - \overline{X})^2}{(n-1)s_X^2}}\end{aligned} \qquad \text{(A2.4)}$$

where $t_{\alpha/2,n-2}$, is the t-statistic (at $100(1 - \alpha)\%$ confidence-level and $(n - 2)$ *dof*). This statistic is obtainable from any statistical tables and σ_{M^*} is the variance of a normally distributed set of residuals around the sample regression line at $X = x^*$. In this way, we can construct a confidence interval for any particular ordinate of interest. Stated in another way, there is now a way to tell how good any $\hat{Y}$ is.

Example
Using the data of Table A2.1, we calculate a 95 percent confidence-interval for the ordinate to the regression line of Figure A2.2 for the household income $x^* = 20$ thousand. First, $\hat{y}^* = 29.33 + 1.15(20) = 52.333$ (which is the same as Y).

$$\sigma_{M^*} \simeq (11.676)\sqrt{\frac{1}{3} + \frac{(20 - 20)^2}{(3 - 1)(10)}} = (11.676)\sqrt{\frac{1}{3}} = 6.741 \qquad \text{(A2.5)}$$

95 percent confidence interval on $E[Y \mid x = 20]$ is therefore $t_{0.025,1}\sigma_{M^*} \cong (12.706)(6.741) = 85.651$. In other words, for the household with the average income of \$20,000, 95 percent of the number of trips made will fall within the band 52.333 ± 85.651. Admittedly, this is a very wide band, reflecting the questionable validity of this toy regression model and the validity of the implicit assumption of having a large number of data points. This is particularly suspect since the band is at its narrowest when $x^* = \overline{X}$, as seen by the calculations in Equation (A2.5). A check on less central positions such as $x^* = 10$ and 30 will verify that the band widths are ±342.60 on either side of the regression line! ■

B. Prediction Interval

Suppose a regression line has been estimated. We obtain another observation $X = x'$, and we are interested in the confidence interval on the estimated Y for this new observation. This typically arises in forecast applications. For instance, future trips (Y') are generated from a target-year population (x'). In other words, the best estimate of $y' = a + bx'$ is to be obtained. The total variance of Y' is made up of two components. The first is the uncertainty of y itself, σ^2, corresponding to the inherent error term ϵ independent of x' in the true regression line $Y' = \alpha' + \beta' \mathrm{x}' + \epsilon$. The second is the statistical estimate on the line that changes with the observations x. As the database pool increases, this second variance term, corresponding to the tilting effect of an additional data point, will go down. Expressed more formally, we have

$$\sigma^2{}_{Y'} = \sigma^2 + \sigma^2{}_{M'} \tag{A2.6}$$

Let us look at this another way. For any given x', the individual values of Y are scattered above and below both the true and the sample regression lines. In practice, it may often be of interest to know how closely one can predict an individual value of Y rather than just the accuracy of the mean value given by the regression line. The formula for a $100(1 - \alpha)\%$ prediction interval for Y is

$$\hat{Y} \pm t_{\alpha/2,\, n-2}\; s_Y \sqrt{1 + \frac{1}{n} + \frac{(x' - \overline{X}\ )^2}{(n-1)s_x^2}}$$

Notice this expression is very similar to Equation A2.4, except that we have identified two components of the variance as shown in Equation A2.6—one corresponding to the inherent uncertainty and the second associated with the additional data point x′.

Example
Continuing the same trip generation example, the 95 percent prediction interval for a single trip observation Y at income $x' = 25$ thousand is

$$58.08 + (12.706)(11.676)\sqrt{1 + \frac{1}{3} + \frac{(25-20)^2}{(3-1)(10)}} = 58.08 \pm 238.432 \tag{A2.7}$$

Again, this toy problem is for illustration only, since the prediction interval is far too huge to be of any use. ■

C. Summary

The entire "interval" problem can be viewed in terms of two graphical illustrations. Figure A2.4 shows the inherent probabilistic element of data. Even if a true regression is obtained by virtue of an infinite number of data points in the sample, there is still a spread of the data, the residuals, around the regression line. In other words, Y is a random variable that follows a probabilistic distribution. The residual, ϵ, is assumed to be normally distributed with a variance of σ^2. In practice, such a true regression line is never obtainable. In its place, an estimated regression line is calibrated based on the model $\hat{Y} = \tilde{a} + \tilde{b}x$, for given values of x. Here $\tilde{a}$ and $\tilde{b}$ are random variables, with specific values of $\tilde{a}^*$ and $\tilde{b}^*$ calibrated by a sample of data points. To predict a value of Y for a given x^* or x' value in practice, two types of errors can be involved: the inherent error σ^2 as discussed above, and the error due to randomness of the regression coefficients themselves, $\sigma^2{}_{\hat{Y}}$ (that is, tilting of the regression line). In other words,

$$\sigma^2{}_Y = \sigma^2 + \sigma^2{}_{\hat{Y}} \tag{A2.8}$$

Figure A2.5 shows that there is randomness in the calibration coefficients a and b, which result in a family of regression lines that are contained in the error

Figure A2.4 PREDICTION BANDS FOR A "TRUE" REGRESSION LINE

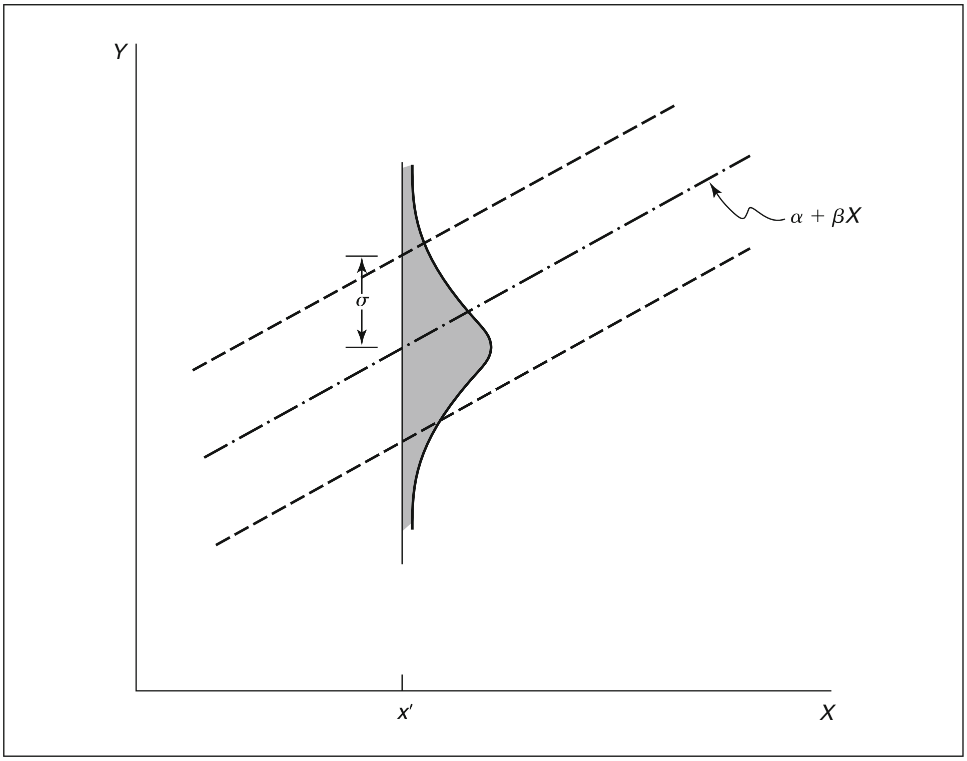
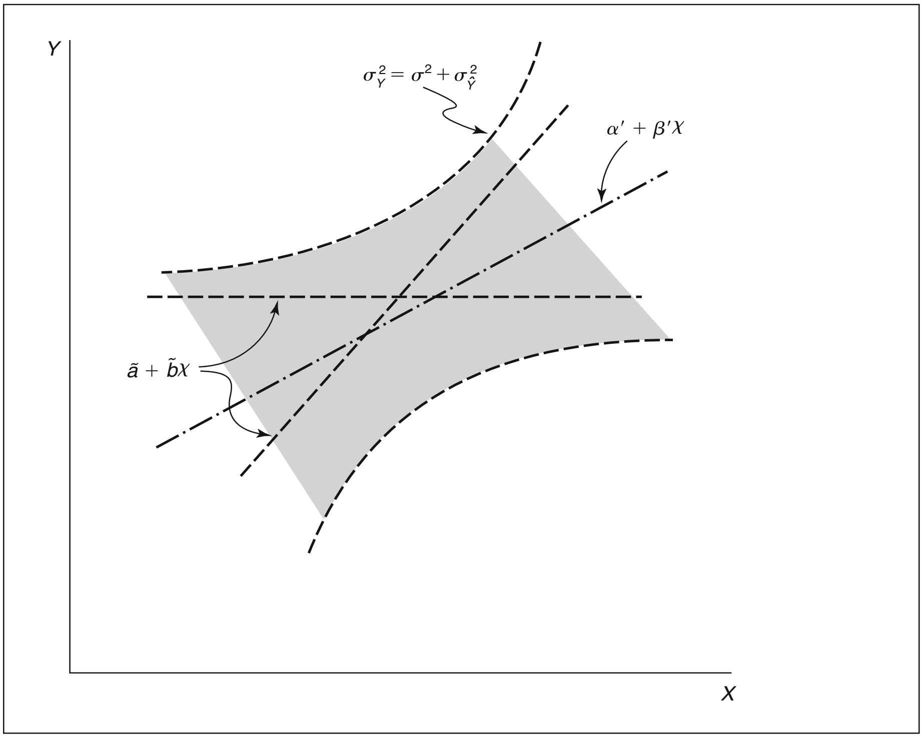


Figure A2.5 CONFIDENCE BANDS FOR AN ESTIMATED REGRESSION LINE



envelope defined by Equation A2.8. This band is a combination of both random and estimation errors. Thus the confidence band is reduced to the prediction band when the data sample is so huge that it encompasses the entire population. In this case the term $\sigma^2{}_{\hat{Y}}$ becomes zero, taking away the curvature of the error envelope and reducing it to a constant band σ^2 around the true regression line as shown in Figure A2.4.

VI. STEPWISE REGRESSION

Stepwise regression is a procedure to search for the best equation automatically. In the case of multiple regression where there are a number of possible explanatory variables (instead of just one as in the toy problem used for illustration so far), it is not at all clear which of them will contribute the most in the regression. Take the following example, which is a trip generation analysis based on stepwise regression for a 29-zone[1] study area (Hutchinson 1974). Here, trips are generated from land use information, including zonal population and employment activities:

$$\begin{aligned} &Y = 69.92 + 1.71X_3 \ \ (R^2 = 0.735, t_3 = 8.7) \\ &Y = 78.63 + 0.78X_2 + 1.24X_3 \ \ (R^2 = 0.888, t_2 = 5.8, t_3 = 8.0) \\ &Y = 58.36 + 1.24X_1 + 0.76X_2 + 0.71X_3 \ \ (R^2 = 0.938, t_1 = 4.7, t_2 = 7.6, t_3 = 4.4) \end{aligned} \tag{A2.9}$$

Here the notation for t-statistics is referenced against the explanatory variable. Thus t_1 is the t-statistic for the first explanatory variable X_1, t_2 the second explanatory variable X_2 and so on. With $t(25 - 27,0.99) \approx 2.8$, the question is: How many of the explanatory variables should be included and which of the three equations is the best?

A. Backward and Forward Regression

To generate these equations, two stepwise regression procedures are commonly found in many statistical packages: the backward elimination procedure and the forward selection procedure (or a combination of both). Specifically, the backward elimination procedure does the following:

1. It includes all variables in the equation to start with.
2. The partial F-test value is then calculated for every variable that is treated as though it were the last variable to enter the equation.
3. The lowest partial F-test value F_L is compared to a pre-selected significance level F_0.
4. If $F_L < F_0$, the variable X_L (which gives rise to F_L) is removed from consideration and the equation re-computed in the remaining variables. Go to step 2.
5. If $F_L > F_0$, adopt the equation as calculated.

The forward selection procedure, on the other hand, performs the following steps:

1. Select the explanatory variable X most correlated with Y (for instance, X_1) and calibrate the equation $Y = f(X_1)$.

2. Find the partial correlation coefficient between X_j ($j \neq 1$) and Y (after allowances for X_1). The X_j with the highest partial correlation coefficient is selected (say X_2) and a second equation $Y = f(X_1, X_2)$ is fitted. Repeat this step until $X_1, X_2, \ldots, X_q$ are in the regression.
3. As each variable is entered into the regression, the following values are examined: R^2 and the partial F-test value for the variable most recently entered. The latter shows whether the variable has taken up a significant amount of the variation over that removed by variables previously in the regression.
4. As soon as the partial F-value related to the most recently entered variable becomes insignificant, the process is terminated.

Example

A regression model for home-based non-work trips in York, Pennsylvania, is to be constructed. Variables are added into the model one at a time. The order of insertion is determined by using the partial correlation coefficient, which measures the importance of a variable not yet in the equation. The selection stops when the contribution of such a variable ceases to be significant at a predetermined level, as measured by the increase in partial F. Table A2.3 summarizes the results of the regression procedure. It can be seen that the variables enter the equation in the order of the largest partial-correlation coefficient (indicating the most significant). The R value increases significantly from the equation with one variable to the one with two variables, indicating a better fit. Adding the third variable to the equation did not change the R value that much, which means that this third variable is

Table A2.3 CALIBRATION RESULTS OF THE FORWARD REGRESSION PROCEDURE

Step		1	2	3
Independent variables		Total employment	Housing land use, Total employment	Housing land use, Total employment, Housing density
New variable		Total employment	Housing land use	Housing density
Partial correlation coefficient of new variables		0.5861	0.5133	0.3290
Partial F-value		20.9306	13.953	4.6139
Multiple correlation coefficient (R)		0.5861	0.71863	0.75422
$\frac{\text{Standard error}}{\text{Average } Y \text{ observations}}$		0.7957	0.6916	0.6616
Analysis of variance	F-ratio (for the whole equation)	20.9306	20.831	16.7121
	Confidence interval[a]	$F(1, 40, 0.99) = 7.31$	$F(2, 39, 0.99) = 5.20$	$F(3, 38, 0.99) = 4.35$

[a] $F(V_1, V_2, 0.99)$ where v_1 is the degree of freedom for the numerator and v_2 for the denominator.

not helping the equation to a better fit. Remember the R value would have a tendency to go up as the number of variables increases (due to a decrease in *dof*). Thus the increase in R^2 can be attributable to statistical properties rather than the explanatory power of a variable.

The partial F-value decreases as more variables are added to the equation, which is normally the case. But the decrease of the partial F-value from the total-employment explanatory variable of the first equation to the housing land-use variable in the second is significant, while that from the second to the third (with the housing-density explanatory variable) is even more significant in comparison. This indicates that by adding the housing land use variable, our confidence level to reject the null hypothesis did not decrease. However, we cannot say the same for the housing density variable. These facts point toward favoring the equation in step 2, resulting in the equation *trip* = *f* (*housing land use, total employment*). This is confirmed by the sharp drop of the overall F-ratio from step 2 to step 3. ■

B. Goodness-of-Fit Parameters for Stepwise Regression

A number of goodness-of-fit parameters are used to measure the significance of the stepwise regression equation. In the backward elimination procedure, the goodness-of-fit parameter used to terminate further deletion of explanatory variables, as pointed out above, is the partial F-value. It is defined as

$$\frac{\textit{variance due to regression with one variable eliminated}}{\textit{variance about regression with none eliminated}}$$

This is the same as

$$\frac{(\text{“}\textit{explained}\text{” } \textit{sum-of-squares})/(k-1)-(\text{“}\textit{explained}\text{” } \textit{sum-of-squares})/[(k-1)-1]}{(\text{“}\textit{unexplained}\text{” } \textit{sum-of-squares})/(n-k)}$$

This indicates the contribution of the coefficient b_i corresponding to the ith independent variable. More precisely, the partial F-value is

$$\frac{\textit{sum-of-squares }(b_i \mid b_{0j}, b_1, \ldots, b_{i+1}, \ldots, b_k)/1}{\textit{error-sum-of-squares}/(n-k)}$$

where the value of the ith variable is the marginal explained sum-of-squares due to the extra degree of freedom. It is, by definition, different from the regular F-value, which is

$$\frac{(\text{“}\textit{explained}\text{” } \textit{sum-of-squares})/(k-1)}{(\text{“}\textit{unexplained}\text{” } \textit{sum-of-squares})/(n-k)}$$

In the forward selection procedure, the square of the partial correlation coefficient is the contribution to explained variation by the candidate variable X_j. It is defined as

$$\frac{\text{“explained”-sum-of-squares (with } X_j \text{ in)} - \text{“explained”-sum-of-squares (with } X_j \text{ out)}}{\text{total-sum-of-squares (with } X_j \text{ out)}}$$

This is the same as

$$\frac{\textit{sum-of-squares}\ (X_j \mid X_1, X_2, \ldots, X_{j-1}, X_{j+1}, \ldots, X_k)}{\textit{total-sum-of-squares}\ (X_1, X_2, \ldots, X_{j-1}, X_{j+1}, \ldots, X_k)}$$

More precisely, it can also be thought of as the percentage of variance in Y not accounted for by other variables, but explained by the variable in question (Kane 1968):

$$\frac{\boldsymbol{R}^2_{Y|X1, X2, \ldots, Xk} - \boldsymbol{R}^2_{Y|X1, \ldots, Xj-1, Xj+1, \ldots, Xk}}{1 - \boldsymbol{R}^2_{Y|X1, X2, \ldots, Xj-1, Xj+1, \ldots, Xk}} \tag{A2.10}$$

Recall that partial correlation describes the extent of linear association that is obtained between a particular pair of variables when other specified variables are held constant. Computationally speaking, it is more convenient to use the following iterative relationship:

$$r_{Y|X_i}(X_j) = \frac{r(Y, X_j) - r(Y, X_j)\, r(X_i, X_j)}{\sqrt{(1 - r^2(Y, X_j))(1 - r^2(X_i, X_j))}} \tag{A2.11}$$

where the notation $r_{Y|X_i}(X_j)$ denotes the partial correlation between Y and X_j given X_i is in the equation already, $r\ (Y, X_i)$ is the unconditional partial correlation of Y with X_i, and $r\ (X_i, X_j)$ is simply the correlation between two independent variables. Should independent variable k be now introduced into the equation, its partial correlation coefficient can simply be calculated from the ones already known:

$$r_{Y|X_iX_j}(X_k) = \frac{r_{Y|X_i}(X_k) - r_{Y|X_i}(X_j)\, r_{Xj|X_i}(X_k)}{\sqrt{(1 - r^2_{Y|X\,i}(X_j))\ (1 - r^2_{Xj|X_i}(X_k))}} \tag{A2.12}$$

Notice any unknown quantity in the right-hand side above can be calculated by Equation A2.11 recognizing that i and j can be interchanged, and the definition of the Y variable is also relative.[2]

Partial Correlation Example

For estimating home-based non-work trips, let us calculate, via Equation A2.11, the partial correlation coefficient for housing land use given total employment is already in the regression equation. Using the notations and the correlation matrix given in Table A2.4, we are interested in

$$r_{Y|X_1}(X_2) = \frac{r(Y, X_2) - r(Y, X_1)\, r(X_1, X_2)}{\sqrt{(1 - r^2(Y, X_1))\,(1 - r^2(X_1, X_2))}}$$

Table A2.4 CORRELATION MATRIX FOR CALCULATING PARTIAL-CORRELATION

	Home-based non-work trips (Y)	**Total employment** (X_1)	**Housing land use** (X_2)	**Housing density** (X_3)
Home-based non-work trips (Y)		0.5861	0.2492	0.1213
Total employment (X_1)			−0.2600	0.0188
Housing land use (X_2)				−0.2610
Housing density (X_3)				

$$= \frac{0.2492 - (0.5861)(-0.2600)}{\sqrt{(1 - (0.5861)^2)(1 - (-0.2600)^2)}} = 0.5133 \qquad \text{(A2.13)}$$

This result agrees with that reported in Table A2.3. It can be seen from Equation A2.12 that second-order partial correlation coefficients, such as that for housing density, involve a lot more calculations than the first-order coefficient calculated above, since they build upon the results of Equation A2.11. ■

The partial F-value for forward regression, similar to the backward regression, is defined as

$$\frac{\textit{variance due to regression with another variable added}}{\textit{variance about regression with existing variables}}$$

or

$$\frac{(\text{“}\textit{explained}\text{” } \textit{sum-of-squares})/((k-1)+1) - (\text{“}\textit{explained}\text{” } \textit{sum-of-squares})/(k-1)}{\text{“}\textit{unexplained}\text{” } \textit{sum-of-squares}/(n-k)}$$

or

$$\frac{\textit{sum-of-squares } (b_{k+1} \mid b_0, b_1, b_2, \ldots, b_k)/1}{\textit{error sum-of-squares}/(n-K)} \qquad \text{(A2.14)}$$

It can be seen, again, that the partial F used in forward-regression is different from the regular F. While there are computational shortcuts in calculating partial F, the central ideas are captured in the above discussions and the following example. ■

Partial *F* Example
Consider the home-based non-work-trip forward-regression $Y = f(X_1, X_2)$ again. Let b_1 be the calibration coefficient for X_1 and b_2 for X_2. While the partial *F*'s are part of the output, it can also be gleaned from the analysis-of-variance tables according to the definition of partial *F* above. Notice the partial *F* for b_2 or due to X_2 is computed in the last column of Table A2.5, second entry from the bottom. Of further interest is the fact that the third entry from the bottom as indicated is not the partial *F* for b_1. The value for b_2 agrees with the partial *F* as documented in Table A2.3. Both are 13.953 in value. ■

It should be noted that the goodness-of-fit parameter, the coefficient of multiple determination (R^2), is not comparable between the various steps of the stepwise regression. The reason is that the (*dof*) change from one step to another. Recall that

$$R^2 = 1 - \; = \frac{\Sigma_i(y_i - \hat{Y})^2}{\Sigma_i\,(y_i - \overline{Y})^2} \qquad \text{(A2.15)}$$

where the numerator has $n - k$ *dof* while the denominator has $n - 1$. As more and more explanatory variables are added, k becomes larger or $n - k$ becomes smaller. There is a tendency for the numerator to become smaller as there is less variability (remember the case of zero *dof* when a regression line is fitted on two points.) One way to compensate for this is to adjust Equation A2.14 as follows

$$\text{Adjusted-}R^2 = \overline{R}^2 = 1 - (1 - R^2)\,\frac{n-1}{n-k}$$

Adjusted-*R*² Example
In the home-based non-work trip regression $Y = f(X_1, X_2)$ shown in Table A2.3, $R^2 = 0.5164$. The adjusted R^2 becomes $R^2 = 1 - (1 - 0.5164)[(42 - 1)/(42 - 3)] = 0.4916$. As expected, $\overline{R}^2$ is smaller than the R^2. Put in another way, R^2 is inflated in comparison to the $\overline{R}^2$ due to the diminished *dof*. ■

Table A2.5 EXAMPLE OF PARTIAL *F* FOR FORWARD REGRESSION[a]

Source of variation	Degree of freedom	Sum of squares	Mean square	*F*-value
Total (corrected)	41	1.95392×10^3		
Due to regression$\lvert b_0$	2	1.00920×10^3	5.04602×10^7	2.08310×10[b]
Due to $b_1 \lvert b_0$	1	6.71205×10^7[c]	6.71205×10^7	2.09306×10
Due to $b_2 \lvert b_1, b_0$	1	3.38000×10^7[d]	3.38000×10^7	1.39531×10[e]
Residual	39	9.44723×10^7	2.42237×10^6	

[a] Unless indicated otherwise, all figures are from the analysis of variance table of $Y = f(X_1, X_2)$.
[b] *F*-value for the entire regression $Y = f(X_1, X_2)$.
[c] From the analysis of variance table of $Y = f(X_1)$.
[d] Computed from equation A3.14.
[e] Partial *F* for b_2 due to X_2.

VII. MATRIX APPROACH TO LINEAR REGRESSION

In the context of multiple linear regression, it is convenient to generalize our parameter estimation discussions in terms of matrix notations. **Y** is defined to be the vector of n observations y_t where $t = 1, 2, \ldots, n$; **X** is the $n \times (k + 1)$ matrix of independent variables constituting the observations x_t; (where $t = 1, 2, \ldots, n$ and $j = 1, 2, \ldots k$; **b** is the vector of parameters to be estimated (consisting of intercept a and coefficients $b_1, b_2, \ldots, b_k$; and $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)$ is the vector of errors. Our bivariate toy problem can be put in matrix notation as follows:

$$\mathbf{Y} = (65, 50, 42)^T, \mathbf{X} = \begin{bmatrix} 1 & 30 \\ 1 & 20 \\ 1 & 10 \end{bmatrix}, \mathbf{b} = (a, b)^T, \epsilon = (\epsilon_1, \epsilon_2, \epsilon_3)^T$$

Now we can write the matrix regression equation as

$$\mathbf{Y} = \mathbf{Xb} + \epsilon \qquad \text{(A2.16)}$$

This is simply a compact way of writing

$$\begin{bmatrix} 65 \\ 50 \\ 42 \end{bmatrix} = \begin{bmatrix} 1 & 30 \\ 1 & 20 \\ 1 & 10 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix} \qquad \text{(A2.17)}$$

which constitutes a simultaneous set of three equations. A set of normal equations can be defined from Equation A2.16 by first writing $\mathbf{Xb} = \mathbf{Y}$ (leaving out the error vector) and then pre-multiplying by $\mathbf{X}^T$, resulting in $\mathbf{X}^T\,\mathbf{Xb} = \mathbf{X}^T\,\mathbf{Y}$. From these normal equations, the coefficients **b** can be solved, yielding the least square estimates (a, b): $\mathbf{b} = (\mathbf{X}^T\,\mathbf{X})^{-1}\mathbf{X}^T\,\mathbf{Y}$.

It can be seen that

$$\mathbf{X}^T\mathbf{X}^{-1} = \begin{bmatrix} 1 & 1 & 1 \\ 30 & 20 & 10 \end{bmatrix} \begin{bmatrix} 1 & 30 \\ 1 & 20 \\ 1 & 10 \end{bmatrix} = \begin{bmatrix} 1+1+1 & 30+20+10 \\ 30+20+10 & 30^2+20^2+10^2 \end{bmatrix} = \begin{bmatrix} n & \Sigma x_t \\ \Sigma x_t & \Sigma x^2{}_t \end{bmatrix} \qquad \text{(A2.18)}$$

It can also be shown that

$$(\mathbf{X}^T\,\mathbf{X})^{-1} = \frac{1}{n\Sigma(x_t - \overline{X})^2} \begin{bmatrix} \Sigma x_t^2 & -\Sigma x_t \\ -\Sigma x_{\mathrm{t}} & n \end{bmatrix} = \begin{bmatrix} 2.333 & -0.1 \\ -0.1 & 0.005 \end{bmatrix} \qquad \text{(A2.19)}$$

In addition,

$$\mathbf{X}^T\,\mathbf{Y} = \begin{bmatrix} 1 & 1 & 1 \\ 30 & 20 & 10 \end{bmatrix} \begin{bmatrix} 65 \\ 50 \\ 42 \end{bmatrix} = \begin{bmatrix} 65+50+42 \\ (30)(65)+(20)(50)+(10)(42) \end{bmatrix} = \begin{bmatrix} \Sigma y_t \\ \Sigma x_t Y_t \end{bmatrix} = \begin{bmatrix} 157 \\ 3370 \end{bmatrix} \qquad \text{(A2.20)}$$

Thus

$$\mathbf{b} = \begin{bmatrix} 2.333 & -0.1 \\ -0.1 & 0.005 \end{bmatrix} \begin{bmatrix} 157 \\ 3370 \end{bmatrix} = \begin{bmatrix} 29.28 \\ 1.15 \end{bmatrix}$$

Within the numerical round-off errors of a basis inversion and the number of significant figures carried, this agrees with previous calculated values of $a = 29.33$ and $b = 1.15$.

VIII. NONLINEAR REGRESSION

Regression models need not be linear. Suppose the postulated model is of the form

$$Y = f(X_1, X_2, \ldots, X_k; \delta_1, \delta_2, \ldots, \delta_r) + \epsilon \tag{A2.21}$$

where δ_j are the estimated parameters (Draper and Smith 1966). If we write $\mathbf{X}^T = (X_1, X_2, \ldots, X_k)$; $\delta^{\mathrm{T}} = (\delta_1, \delta_2, \ldots, \delta_r)$, Equation A2.21 can be rewritten compactly as

$$Y = f(\mathbf{X}, \delta) + \epsilon \tag{A2.22}$$

Notice that k is not necessarily the same as r—that the number of estimated parameters r do not necessarily have to be equal to the number of independent variables k in general. We shall assume that errors (ϵ) are uncorrelated, that $\mathrm{var}(\epsilon) = \sigma^2$, ϵ is independent and normally distributed with a mean of zero and variance σ^2.

When there are n observations of the form $\mathrm{Y}_{\mathrm{t}}, X_{1t}, X_{2t}, \ldots, X_{kt}$ for $t = 1, 2, \ldots, n$, we can write the model (22) as

$$Y_t = f(\mathbf{X}_t, \boldsymbol{\delta}) + \epsilon_t \quad t = 1, 2, \ldots, n \tag{A2.23}$$

where $\mathbf{X}_{t\,=} (X_{1t}, X_{2t}, \ldots, X_{kt})^T$. The assumption of normality and independence of the errors can now be written compactly as $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{I}\sigma^2)$ where $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)^T$, and as usual $\mathbf{0}$ is a vector of zeros and $\mathbf{I}$ is an identity matrix. We define the error sum-of-squares for the nonlinear model and the given data as

$$S(\boldsymbol{\delta}) = \sum_{t=1}^{n} [y_t - f(\mathbf{X}_t, \boldsymbol{\delta}_0)]^2 \tag{A2.24}$$

The above is also referred to as the sum-of-squares surface. Notice that since y_t and $\mathbf{X}_t$ are fixed observations, the sum of squares is simply a function of $\boldsymbol{\delta}$. The $\boldsymbol{\delta}$ so obtained is referred to as the conditional estimate, in the sense that they are conditioned upon the given values of y_t and $\mathbf{X}_t$. We shall denote by $\hat{\boldsymbol{\delta}}$ a least squares estimate of $\boldsymbol{\delta}$—a value of $\boldsymbol{\delta}$ that minimizes $S(\boldsymbol{\delta})$. It can be shown that under the conditional assumptions, the least squares estimate of is also the maximum likelihood estimate.[3] This is because the likelihood function for this problem can be written as $L(\boldsymbol{\delta}, \sigma^2) = (2\pi\sigma^2)^{-n/2} \exp\,[-\,S(\boldsymbol{\delta})/2\sigma^2]$, so that if σ^2 is known, maximizing $L(\boldsymbol{\delta}, \sigma^2)$ with respect to δ is equivalent to minimizing $S(\boldsymbol{\delta})$ with respect to $\boldsymbol{\delta}$.

To find the least squares estimate $\hat{\boldsymbol{\delta}}$, we need to differentiate Equation A2.24 with respect to $\hat{\boldsymbol{\delta}}$. This provides the r normal equations, which must be solved for $\hat{\boldsymbol{\delta}}$:

$$\sum_{t=1}^{n} \left\{ [y_t - f(\mathbf{X}_t, \delta)] \left[\frac{\partial f(\mathbf{X}_{t,} \boldsymbol{\delta})}{\partial \boldsymbol{\delta}_i} \right] \right\}_{\boldsymbol{\delta} = \hat{\boldsymbol{\delta}}} = 0 \quad i = 1, 2, \ldots, r \qquad \text{(A2.25)}$$

Notice the derivative in square brackets is evaluated at the corresponding estimated values $\hat{\boldsymbol{\delta}}$, which have the same subscript. When the function $f(\mathbf{X}_t, \delta)$ is linear

$$f(\mathbf{X}_{t,} \boldsymbol{\delta}) = \delta_1 X_{1t} + \delta_2 X_{2t} + \ldots + \delta_r \mathbf{X}_{rt} \qquad \text{(A2.26)}$$

this derivative is a function of the $\mathbf{X}_t$ only: $\partial f/\partial\delta_i = X_{it}$ for $i = 1, 2, \ldots, r$ and does not involve δ at all. This leaves the normal equations in the form of linear equations in $\boldsymbol{\delta}$ as discussed in the previous section.

$$\sum_{t=1}^{n} [y_t - f(\mathbf{x}_{t,} \hat{\boldsymbol{\delta}})]\, x_{it} = \sum_{t=1}^{n} (y_t - \hat{y}_t)\, x_{it} = 0 \qquad i = 1, 2, \ldots, r \qquad \text{(A2.27)}$$

which is equivalent to $\mathbf{X}^T (\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\delta}}) = \mathbf{0}$, where the estimated parameters $\hat{\boldsymbol{\delta}}$ is the same as $\mathbf{b}$ in linear regression. While this is a simple set of equations to solve for linear regression, it is quite complicated for nonlinear cases, as demonstrated by the example below. Unfortunately, nonlinear regression is the rule rather than the exception in spatial time series, as demonstrated in the "Space Time Modeling" section of the "Spatial Time Series" chapter of Chan (2005).

Example

Suppose we wish to find the normal equation(s) for obtaining the least squares estimate $\hat{\boldsymbol{\delta}}$ *of* $\boldsymbol{\delta}$ for the model $Y = f(\delta, \tau) + \epsilon$ where $f(\delta,\tau) = \exp(-\delta\tau)$ and where n pairs of observations $(y_1, t_1), (y_2, t_2), \ldots, (y_n, t_n)$ are available. We take the derivative $\partial f/\partial\delta = -\,\tau e^{-\delta\tau}$. Applying Equation A2.25 yields a single normal equation

$$\sum_{j=1}^{n} [y_j - \exp(-\hat{\boldsymbol{\delta}} t_j)]\,[-t_j \exp(-\hat{\boldsymbol{\delta}} t_j)] = 0 \qquad \text{(A2.28)}$$

We can see that even with one parameter and a comparatively simple nonlinear model, finding $\hat{\delta}$ by solving the only normal equation is not easy. When more parameters are involved and the model is more complicated, the solution of the normal equations can be extremely difficult to obtain; iterative methods must be employed in nearly all cases. To compound the difficulties, multiple solutions may exist, corresponding to multiple stationary values of the surface function $S(\hat{\boldsymbol{\delta}})$. ■

IX. CONCLUDING REMARKS

This appendix reviewed some of the basic statistical concepts. We discussed estimators in their single, two-dimensional, and multidimensional contexts, covering linear regression, stepwise regression, and nonlinear regression. Aside from the current volume, this introduction is meant to supplement such other chapters as "Generation, Competition and Distribution" and "Spatial Time Series" in Chan (2005), with pertinent fundamental principles. It does not pretend to replace excellent texts such as those listed in the references. However, care has been taken to make the presentation as self-contained and tutorial as possible, and to include as many examples as necessary—conducted sometimes at the sacrifice of mathematical rigor. The focus is on model calibration.

In linear regression, we pointed out that no single statistic can by itself speak for the overall quality of the regression equation. A number of statistics need to be considered together in assessing the quality of a calibration. Goodness-of-fit parameters are confirmations of a priori professional judgment on the hypothesized structural equation. The ultimate quality of a regression equation depends critically on the judicious choice of a sound structural equation, not just on statistical tests. Hypothesis about such a regression equation should therefore be re-examined all the time during a regression exercise to ensure a sound model.

Stepwise regression procedures are by no means perfect ways to automate the selection of a best equation. They are heuristic ways based on selective statistics such as partial-F or partial correlation coefficient. As pointed out in the text, the values of partial-F and partial correlation coefficients are conditioned on what variables are already included in the equation. A different sequence with which explanatory variables are introduced in the equation will result in different values for the same coefficient. Thus the use of partial-correlation coefficients in an iterative application of forward and backward regression fails to test whether the elimination of a variable, for instance, X_j might have made it impossible for readmission.

In practical applications, it is desirable to have a parsimonious equation, meaning a simple model that is statistically significant. Too many explanatory variables will not only introduce the problems associated with a diminishing degree of freedom, it will often pose prohibitive data-collection requirements. Let us replace the first equation in Equation 9 with $Y = 50.02 + 0.84X_4$ ($R^2 = 0.93$, $t_4 = 19.2$). The goodness-of-fit statistics are similar between the first and third regression equations. Both are superior to the second equation; however, since the first equation is more parsimonious, it is preferred to the third. It can be seen therefore that the selection of the best equation is not a purely statistical exercise. It requires the combination of statistical tests with professional judgement in an artful manner.

Matrix notation for regression is convenient, particularly in the case of multiple regression and nonlinear regression. Through such a notation, one can easily see the relationship, say, between linear and nonlinear calibration. It was shown that while calibration of linear regression is relatively straightforward, calibration of nonlinear regression is not. One of the state-of-the-art issues in this field is the availability of stable and accurate nonlinear calibration procedures (Seber and Wild 1989). Unfortunately, the real world is replete with examples of nonlinear models. In spite of valiant attempts to provide generalized calibration tools, practical computational experiences tend to be case-specific, as we can see in the main body of this text.

ENDNOTES

[1] A zone (or more accurately traffic zone) is a subregion of a study area, each of which has its zonal attributes such as population and employment.

[2] This procedure can be generalized to an autoregressive time-series, as explained in the "Estimating the Parameters" subsection of the "Spatial Time-Series" Chapter in Chan (2005).

[3] The maximum-likelihood estimation procedure is explained in Chapter 3.

REFERENCES

Chan, Y. (2005). *Location, transport and land-use: Modelling spatial-temporal information.* Berlin and New York: Springer.

Crow, E. L.; Davis, F. A.; Maxfield, M. W. (1960). *Statistics manual: With examples taken from ordinance development*. New York: Dover.

Draper, N. R.; Smith, H. (1966). *Applied regression analysis.* New York: Wiley

Freeman, H. (1963). *Introduction to statistical inference.* Reading, Massachusetts: Addison-Wesley

Hutchinson, B. G. (1974). *Principles of urban transport systems planning.* New York: McGraw-Hill.

Kane, E. J. (1968). *Economic statistics and econometrics: An introduction to quantitative economics.* New York: Harper and Row.

Seber, G. A. F.; Wild, C. J. (1989). *Nonlinear regression*. New York: Wiley.

Wallace, T. D.; Silver, J. (1988). *Econometrics: An introduction.* Reading, Massachusetts: Addison-Wesley.

Appendix 3

Review of Pertinent Markovian Processes

In making locational decisions over time, we face a world of uncertainty. In this appendix, we put together the basic concepts behind time-dependent probabilistic processes (or stochastic processes), including Poisson (random), queuing, Markov, and state-transition procedures in general. Particularly of interest is optimizing a Markovian decision system. A basic building block of these methodologies is the Markovian (or memoryless) property, which suggests independence among sequential outcomes. By discussing the "memoryless properties" that govern many of these phenomena, reference is made to dynamic programming, non-Markovian processes and compartmental models as well. All these are extensions to the basic concepts. While compartmental models will be discussed in detail in Appendix 1, their relationship to Markov process is delineated here. The probability self-instructional model serves as an excellent introduction to this appendix.

I. POISSON PROCESS

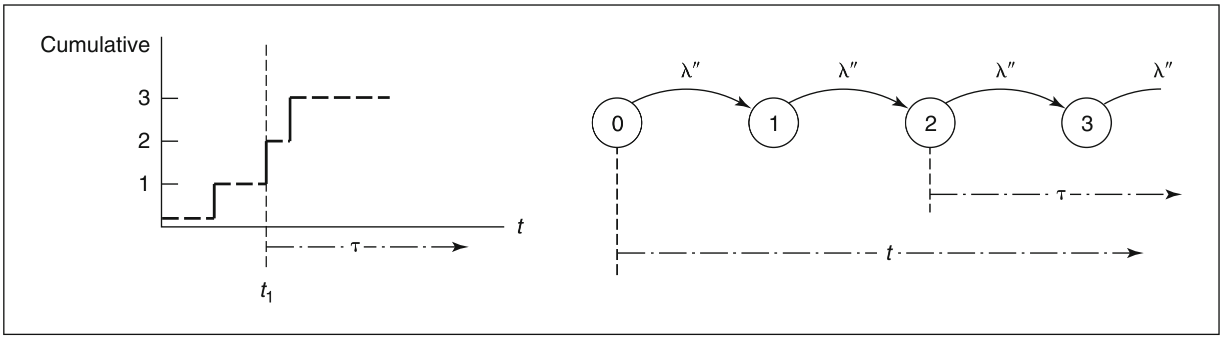
One of the motivations to study stochastic process is to address problems of congestion. Congestion is often manifested in terms of waiting lines (or queues) at a service facility such as a fire station, which has to respond to probabilistic demands over the entire neighborhood it serves. Given demands are usually random and there is a limited number of fire engines, the fire station can be taxed to its limit on occasions. Stochastic process helps us to understand such situations and to offer possible solutions.

A. State Transition Equations

The first step in the analysis process is to understand how random demands arrive. To represent random demands, consider the state transition diagram shown in Figure A3.1, where each state stands for the number of demands arriving in the period of time *t*. Let p_{ij} stand for the probability of transitioning from state *i* to *j* or that the demand changes from *i* to *j*. The differential equations governing the evolution of the system over time, when demands arrive at an average rate of λ'', becomes

$$\begin{aligned}
&\dot{p}_{00}(t) = -\lambda'' p_{00}(t) \\
&\dot{p}_{01}(t) = \lambda'' p_{00}(t) - \lambda'' p_{01}(t) \qquad \text{(A3.1)} \\
&\dot{p}_{02}(t) = \lambda'' p_{01}(t) - \lambda'' p_{02}(t) \\
&. \\
&. \\
&. \\
&\text{etc.,}
\end{aligned}$$

Figure A3.1 CUMULATIVE ARRIVAL PATTERN AND ASSOCIATED STATE-TRANSITION DIAGRAM



or $\dot{p}_{0k}(t) = \lambda'' p_{0k-1}(t) - \lambda'' p_{0k}(t)$ for $k = 0, 1, \ldots, n$; where $\dot{p}_{0i}{}'(t)$ is the time derivative of the probability of transitioning to state *i.* The subscript 0 simply suggests the system starts empty. In matrix form,

$$\begin{bmatrix} \dot{p}_{00}(t) \\ \dot{p}_{01}(t) \\ \dot{p}_{02}(t) \\ \cdot \\ \cdot \\ \cdot \\ \dot{p}_{0n}(t) \end{bmatrix} = \begin{bmatrix} -\lambda'' & & & & \\ \lambda'' & -\lambda'' & & & \\ & \lambda' & -\lambda'' & & \\ & & & \cdots & \\ & & & & \\ & & & & \\ & & & & \cdots \end{bmatrix} \begin{bmatrix} p_{00}(t) \\ p_{01}(t) \\ p_{02}(t) \\ \cdot \\ \cdot \\ \cdot \\ p_{0n}(t) \end{bmatrix} \qquad \text{(A3.2)}$$

or $\dot{\mathbf{p}}_0(t) = \mathbf{\Pi}\,\mathbf{p}_0(t)$, which describes the system evolving over time increments *dt.* Here $\mathbf{\Pi}$ is the matrix of transition rates from state 0 to state *x.*

Figure A3.2 INTERARRIVAL TIME DISTRIBUTION FUNCTION

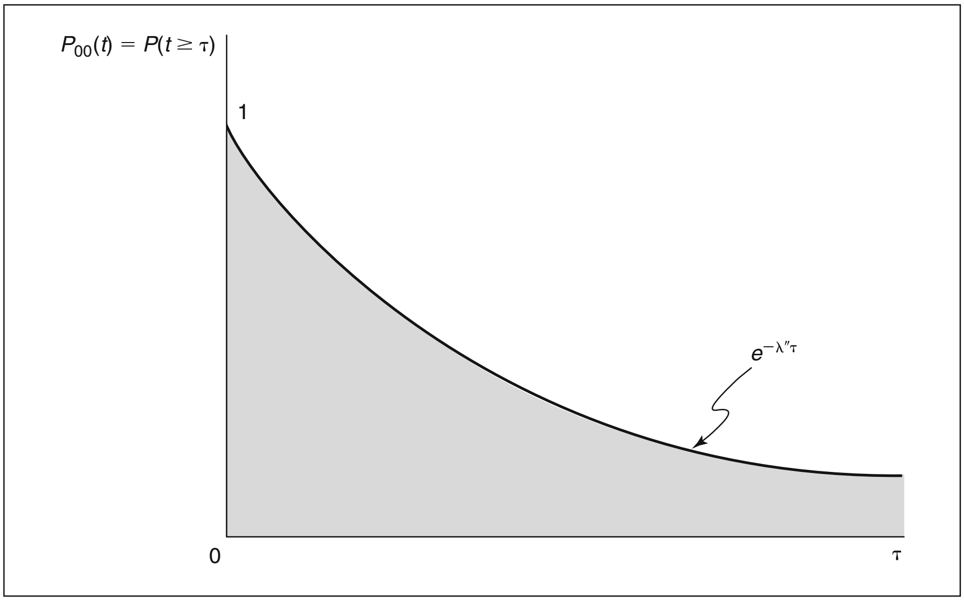
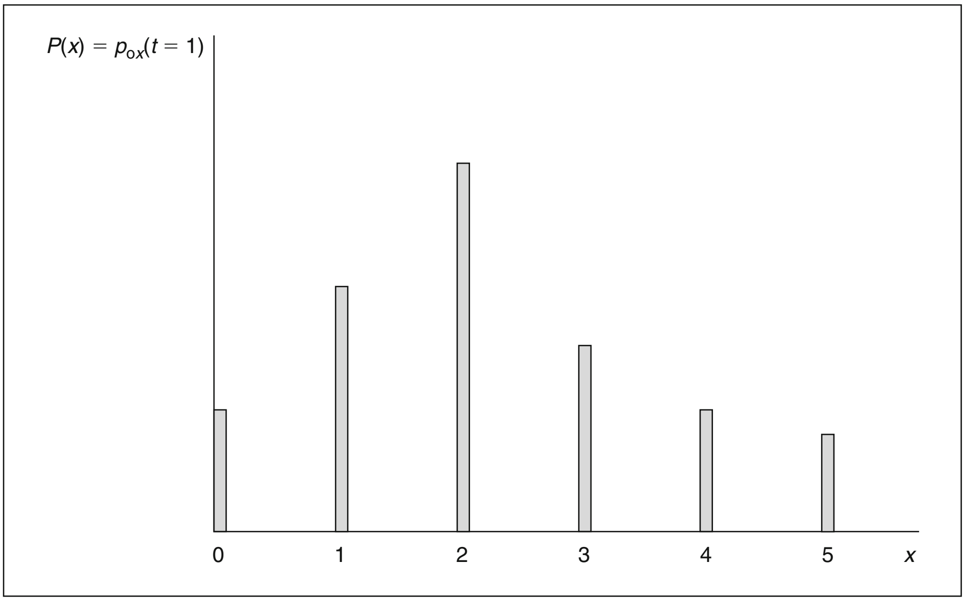


Figure A3.3 POISSON-ARRIVAL DISTRIBUTION FUNCTION



B. Solution to Random Process

Solving the first line of Equation A3.1 by integration, with the constant of integration $p_{00}(0) = 1$, we have $p_{00}(t) = e^{-\lambda''\tau}$. This curve is plotted in Figure A3.2 for illustration. Notice this is the probability that there is no event in time τ, or the interarrival time t is greater than τ: $P(t \geq \tau)$. Substituting this integration result into the second equation and integrating again yields $p_{01}(t) = e^{-\lambda''t}$. Repeating the process one more time for the next equation, we have: $p_{02}(t) = (\lambda''t)^2 e^{-\lambda''t}/2$. In general, $p_{0x}(t) = (\lambda''t)^x e^{-\lambda''t}/x!$, for $x = 0, 1, 2, \ldots, n$. Thus for a given time period $t = \tau$, the vector $\mathbf{p}_0$ gives the probability distribution of the various states, and the sum of the entries in the vector is unity by definition of a probability distribution. Such a Poisson distribution is plotted in Figure A3.3. When τ is normalized to one time unit, we have the alternate expression for Poisson distribution:

$$P(x) = \frac{\lambda''e^{-\lambda''}}{x!} \qquad x = 0, 1, 2, \ldots, n. \tag{A3.3}$$

II. FIELD DATA FROM AIR TERMINAL

The somewhat abstract ideas above can be illustrated with concrete data (Morlok 1978). Suppose we collected the information of Table A3.1 at an air terminal during an eight-hour day. Here in this figure one time unit represents a half hour. For the time being, we only examine the first column—arrival time of an aircraft—since we are interested in the demands placed upon the terminal. In this case, the first aircraft arrives 1/2 hour after the day begins, the second one arrives one hour after the day starts and so on. The average arrival rate λ" is computed as 12/16 or 0.75 vehicles per unit time.

Table A3.1 AIRCRAFT ARRIVAL AND DEPARTURE FIELD DATA

Aircraft	Arrival time*	Depart time*	Wait time	Total time in system	Service time	Total in system	Interarrival time
#1	1	2	0	1	1	1	—
2	2	3	0	1	1	1	1
3	3	4	0	1	1	1	1
4	5	6	0	1	1	1	2
5	8	9	0	1	1	1	3
6	8.1	10	0.9	1.9	1	2	0.1
7	9	11	1	2	1	2	0.9
8	9.5	12	1.5	2.5	1	3	0.5
9	11	13	1	2	1	2	1.5
10	12.5	14	0.5	1.5	1	2	1.5
11	14	15	0	1	1	1	1.5
12	16	17	0	1	1	1	2

*Data to be collected, rest can be derived.

SOURCE: Morlok (1978). Reprinted with permission.

A. Exponential Distribution

From the discussions in Section I-B, if arrivals are random, interarrival times are exponentially distributed: $P(t \geq \tau) = e^{-0.75\tau}$ for $0 \leq \tau \leq \infty$, where $P(t \geq \tau)$ is the probability that the interarrival time t is greater than τ. Now we can compare the field data with a theoretical exponential interarrival time distribution. If the two match, then we can conclude that interarrivals are truly random. Table A3.2 shows how this can be conducted. For example, in the first row of the second column, there are clearly the entire 11 (12-1) interarrival times that are larger than 0 unit in duration, considering that an interarrival time is defined for each pair of aircraft. In the second row, we counted only eight interarrival-times that are one unit or longer and so on. Now we plot the theoretical curve against the experimental curve in Figure A3.4, which allows for a visual inspection of the two curves side by side. Notice the average interarrival time, $1/\lambda''$ or 1.333 units, is also graphed in Figure A3.4 for reference.

B. Poisson Distribution

Again from Section I-B, if arrivals are random, the number of aircraft arriving in a unit of time (1/2 hour) constitutes a Poisson distribution: $P(X = x) = (e^{-[0.75]x})/x!$ for $x = 0, 1, 2, 3, \ldots,$ where $P(X = x)$ is the probability that x aircraft arrive in the time unit. Table A3.3 shows both theoretical and field data side by side. For example, there are six occurrences in which no aircraft arrive in a time unit, and eight occurrences in which one aircraft arrives in a time unit and so on. All these come from the first data column of Table A3.1. Comparison between theoretical

Table A3.2 THEORETICAL AND FIELD DATA ON INTERARRIVAL TIME DISTRIBUTION

Experimental			**Theoretical**
Time intervals t	No. interarrival times that exceed the time interval $t > \tau$	Frequency distribution	$P(t > \tau) = e^{-0.75\tau}$
0	11	1.00	1.00
1	8	0.73	0.47
2	3	0.27	0.22
3	1	0.09	0.11
4	0	0	0.05
5	0	0	0.02
6	0	0	0.01
7	0	0	0.01
8	0	0	0

Figure A3.4 INTERARRIVAL TIME DISTRIBUTIONS

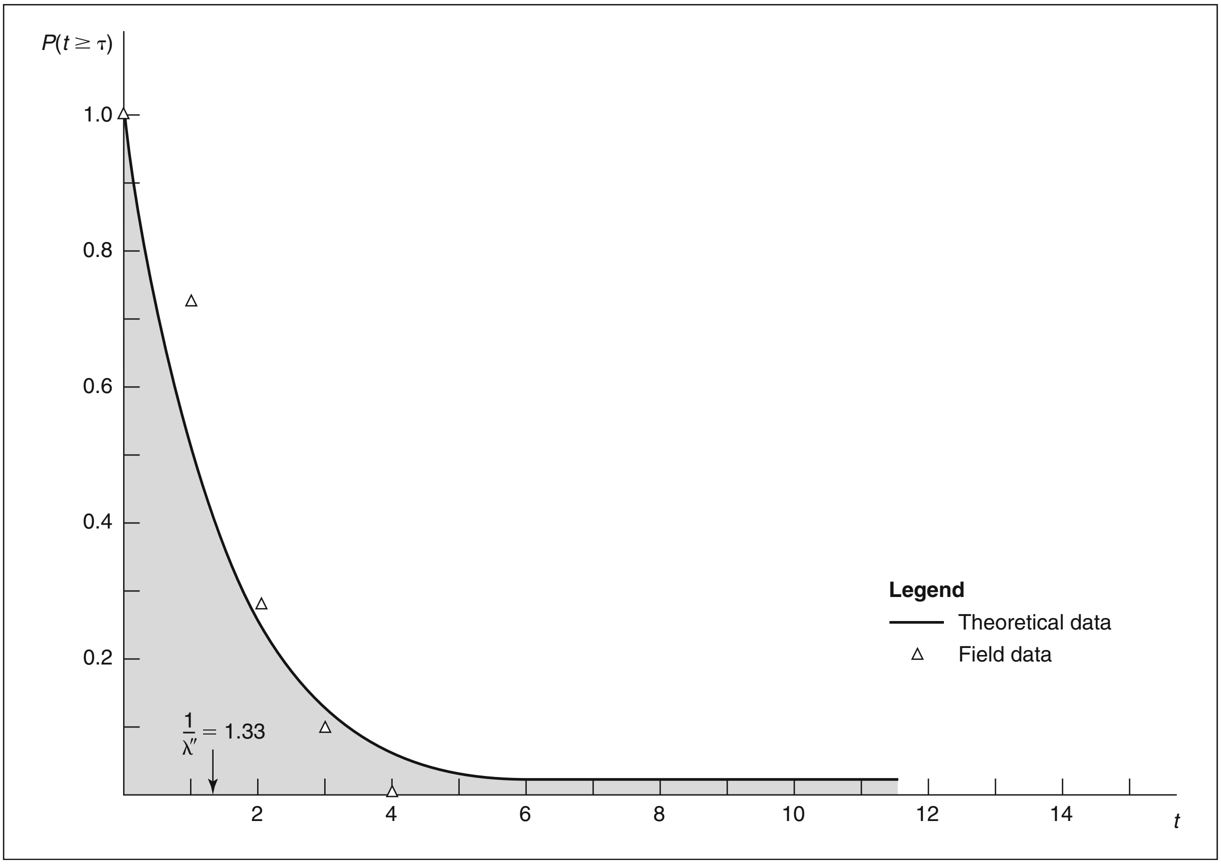
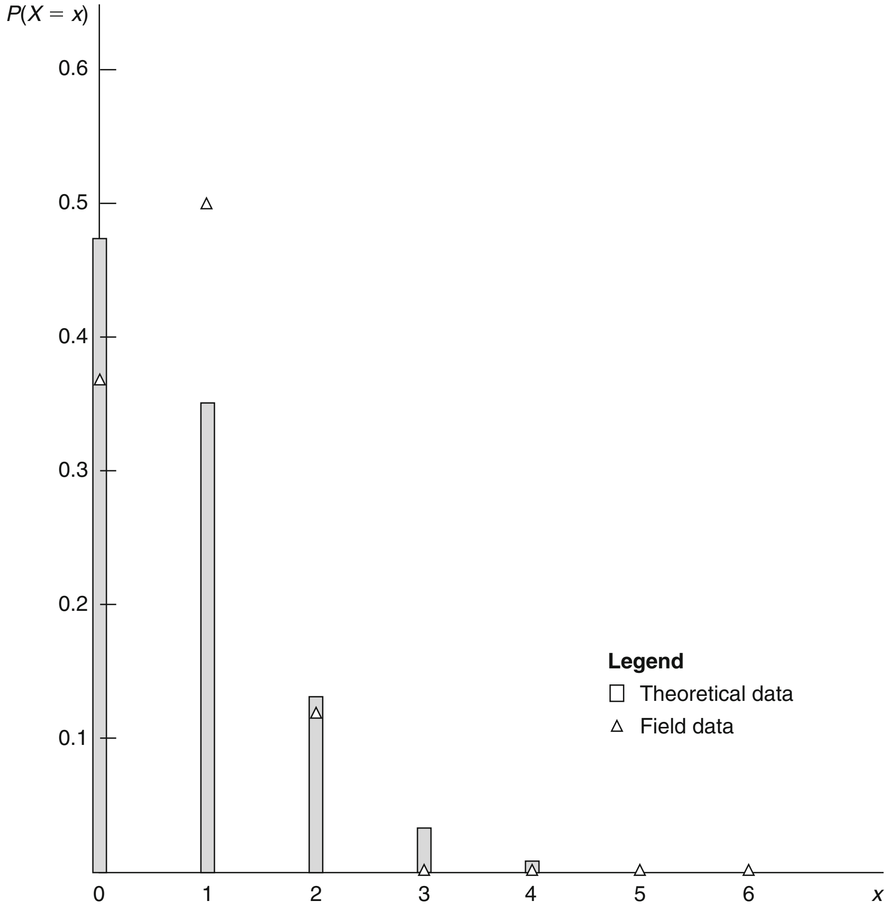


Table A3.3 THEORETICAL AND FIELD DATA ON POISSON DISTRIBUTION

Experimental			Theoretical
No. of arrivals per unit time x	No. of occurrences (time intervals)	Frequency distribution	$P(X = x) = \dfrac{e^{-0.75}[0.75]^x}{x!}$
0	6	0.375	0.47
1	8	0.500	0.35
2	2	0.125	0.13
3	0	0	0.03
4	0	0	0.01
5	0	0	0

Figure A3.5 POISSON DISTRIBUTIONS



and empirical curves in Figure A3.5 does not seem to support the assumption of a random process, even though the previous test using exponential function resulted in a more positive visual verification. Rigorous, scientific goodness-of-fit statistics such as the chi-square should be used in lieu of a manual, visual process.[1]

III. M/M/1 QUEUE

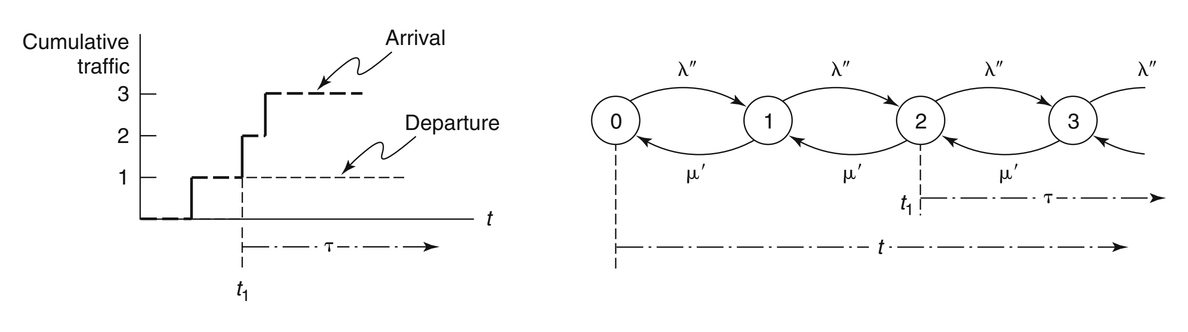
Instead of just the demand arrival pattern, one can use similar state transition equations to derive the full set of waiting-line or queuing equations. This covers both arrivals and departures after receiving service at the terminal. We assume random arrivals and random service here, or more specifically Poisson arrivals with an average rate of λ'' and exponential service time averaging $1/\mu'$. Let $P_i(t)$ be the probability that the system is in state i at time t. In Figure A3.6, we have the transition diagram to describe random arrivals and random service at a single server. The usual convention is to use $M/M/1$ designation where the first M stands for random arrival, the second M stands for random service, and 1 stands for a single server.

Starting with an empty system, or state $i = 0$, the transition differential-equation set is

$$\begin{aligned} \dot{P}_0(t) &= -\lambda''P_0(t) + \mu'P_1(t) \\ \dot{P}_1(t) &= -(\lambda'' + \mu')P_1(t) + \lambda''P_0(t) + \mu'P_2(t) \\ \dot{P}_2(t) &= -(\lambda'' + \mu')P_2(t) + \lambda''P_1(t) + \mu'P_3(t) \\ &\;\;. \\ &\;\;. \\ &\;\;. \\ &\text{etc.} \end{aligned} \qquad \text{(A3.4)}$$

where $\dot{P}_i(t)$ is the time derivative of the probability of being in state i. In general, $\dot{P}_{0i}(t) = -(\lambda'' + \mu')P_i(t) + \lambda''P_{i-1}(t) + \mu'P_{i+1}(t)$ for $i = 0, \ldots, \infty$. Expressed in matrix form:

Figure A3.6 CUMULATIVE ARRIVAL AND DEPARTURE CURVES AND ASSOCIATED TRANSITION DIAGRAM



$$\begin{bmatrix} \dot{P}_0(t) \\ \dot{P}_1(t) \\ \dot{P}_2(t) \\ \cdot \\ \cdot \\ \cdot \\ \dot{P}_n(t) \end{bmatrix} = \begin{bmatrix} -\lambda'' & \mu' & & & \\ \lambda'' & -(\lambda''+\mu') & \mu' & & \\ & \lambda' & -(\lambda''+\mu') & \mu' & \\ & & \cdots & & \\ & & & \cdots & \\ & & & & \cdots \\ & & & & \end{bmatrix} \begin{bmatrix} P_0(t) \\ P_1(t) \\ P_2(t) \\ \cdot \\ \cdot \\ \cdot \\ P_n(t) \end{bmatrix} \qquad \text{(A3.5)}$$

or $\dot{\mathbf{P}}(t) = \widetilde{\Pi}\ \mathbf{P}(t)$, where $0 \le t \le \infty$, and the system evolves over the time increments dt.

For the steady-state (or average) situation, all derivatives are zero, and we have the following equation set after dropping the t argument:

$$\begin{aligned} &\lambda'' P_0 = \mu' P_1 \\ &(\lambda'' + \mu')P_1 = \lambda'' P_0 + \mu' P_2 \\ &(\lambda'' + \mu')P_2 = \lambda'' P_1 + \mu' P_3 \\ &\cdot \\ &\cdot \\ &\cdot \\ &\text{etc.} \end{aligned}$$

Solving these equations yields $P_1 = \rho'_0$, where $\rho' = \lambda''/\mu'$; $P_2 = \rho'^2 P_0$; $P_3 = \rho'^3 P_0$; ... etc. Substituting into the relationship that $P_0 + P_1 + P_2 + P_3 + \ldots = 1$, we have $P_0(1 + \rho' + \rho'^2 + \ldots\ ) = 1$, or $P_0 = (1 - \rho')$. Now, re-substituting back $P_1 = \rho'(1 - \rho')$, $P_2 = \rho'^2(1 - \rho')$, $P_3 = \rho'^3(1 - \rho')$, ... and so forth. The average length of a waiting line or queue, including the one being served, is therefore

$$\begin{aligned} \overline{L} &= 0P_0 + 1P_1 + 2P_2 + \ldots \\ &= (1 - \rho')(\rho' + 2\rho'^2 + 3\rho'^3 + \ldots) \\ &= (1 - \rho')\rho'(1 + 2\rho' + 3\rho'^2 + \ldots) \\ &= (1 - \rho')\rho'(1 - \rho')^{-2} = \rho'/(1 - \rho') \end{aligned}$$

From the queue length, other queuing statistics can be derived. For example, the total time in the system, which is the amount of time for the last arrival to spend in line plus the time being served is simply $(1/\mu')\overline{L}$, or $= \rho'/\mu'(1 - \rho')$.

IV. QUEUING SYSTEMS

The above derivations are based on a set of state transition equations that describe a Markov process—especially, a continuous-time Markov process. Similar processes can be used to model other types of queues. For example, if we have established that the arrivals are not random in the air terminal example, some other distributions may fit the data better, and the $M/M/1$ queue may not be an appropriate model to use in this case. We wish to present several queuing models below. But due to space limitation, we will not show the detailed steps of derivation, as we have done in the case of $M/M/1$ queue. Interested readers should refer to standard texts on queuing for details (See Cooper [1980] for example).

A. Basic Theory

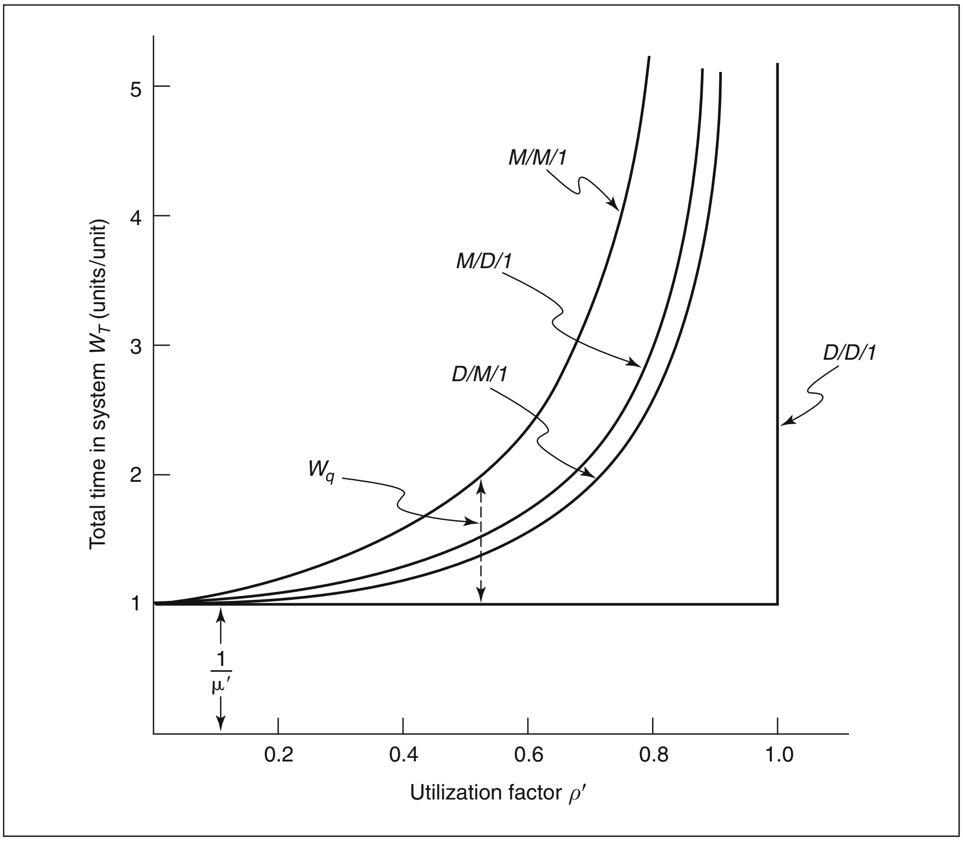
The basic idea behind queuing is really quite straightforward. We have a stream of demands coming in, and they are being met by a service facility. The demand traffic eventually exits after being served at the end of the process. A schematic describing this phenomenon can be sketched: $\lambda'' \rightarrow \mu' \rightarrow \ldots$ where in the air terminal example both parameters $\lambda''$and μ' are measured in vehicles/time-unit. λ'' is called the average rate of arrival, and μ' is the average rate of service. As defined in Section III, λ''/μ' is called the utilization factor ρ', or the traffic intensity, signifying the percentage of time the server is busy on the average. Broadly speaking, there are two types of queuing: deterministic and probabilistic. A deterministic queue is straightforward; it is analogous to a sink with a running faucet. Water enters the sink via the faucet at a precise rate of λ'', and the sink drains at a precise rate of μ'. Unless the water comes in faster than going out, or $\lambda'' > \mu'$, there is no water backup, which is analogous to saying that no queue is formed. When $\lambda'' > \mu'$, water backs up in the sink and the water level keeps on rising, resulting in a wet floor when water eventually overflows. In the case of a probabilistic process, a queue may be formed even though that on the average $\lambda'' < \mu'$, since the water is coming in and going out at fluctuating rates. Thus on occasions, the water gushes out of the faucet while the drain is sluggish, causing water backup, even though on the average, the sink is supposed to drain faster than the incoming rate at the faucet.

We can summarize the probabilisic situation with the following table:

λ''	μ'	delay W_q
random	random	worst
random	constant	medium
constant	random	least

which says that if the faucet runs randomly, and the drain works randomly, the water backs up and the water in the sink takes a long time to clear on the average. If the faucet runs randomly, but the drain is perfectly reliable, the situation is more under control. The best is when the incoming water is steady, even though the drain may be haphazard. All these refer to the situation when $\lambda'' < \mu'$. Obviously, we are guaranteed an infinite backup and a wet floor when $\lambda'' \geq \mu'$ to begin with. Standardized short-hand notations for random is M (as mentioned previously) and for constant D. Based on this notation, the queuing system above in the second line of the table is an $M/D/1$ model, where the last number again denotes one single server, similar to the case of $M/M/1$ queue. The average system-behavior is summarized in Figure A3.7, which shows the steady-state (or stationary) behavior of the queues. On the average, the delay is at its worst for $M/M/1$ queue, and least for $D/D/1$ (until the water spills over beginning at $\lambda'' \geq \mu'$). Notice the figure is dimensionless, in that both scales of the horizontal and vertical axes are independent of any particular unit of measurement. First of all, the utilization factor is clearly dimensionless. Total time in the system (in units/unit) is scaled with the average service time being 1 unit. In our air terminal example, it simply means that everything is a multiple of 1/2 hour. The total time W_T is the sum of the delay time in queue W_q and the service time $1/\mu'$. It refers to the time spent by a single vehicle unit to be served at the terminal.

Figure A3.7 AVERAGE PERFORMANCE OF VARIOUS QUEUING DISCIPLINES



In the case of multiple servers, a schematic can be drawn as follows:

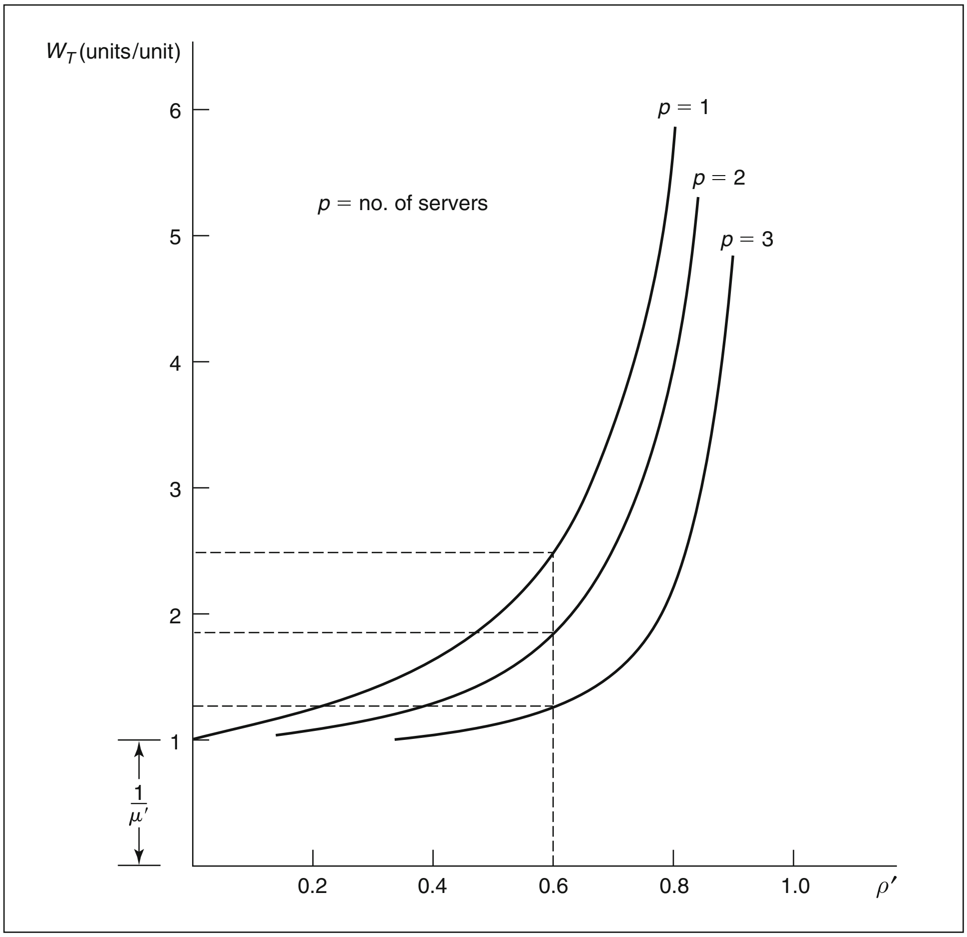
$$\lambda'' \rightarrow \begin{cases} \mu' \rightarrow \\ \mu' \rightarrow \\ \cdot \\ \cdot \\ \cdot \\ \mu' \rightarrow \end{cases} \tag{A3.6}$$

The more servers, the less the delay time, as illustrated by $M/M/p$, or the p-server system shown in Figure A3.8. For example, for a utilization factor $\rho' = 0.6$, the single-server queue incurs the largest system delay (at 2.5 units/unit), the two-server queue less (at 1.8 units/unit) and the three-server queue the least (at 1.3 units/unit).

B. Queuing Formulas

Now back to the single-server system. For a first-come-first-served (FIFO) system, the following queuing equations can be obtained:

Figure A3.8 PERFORMANCE OF MULTI-SERVER QUEUES



queue discipline	queue length (L_q)	delay (W_q)
$D/D/1$	0	0
$D/M/1$	(Intractable analytically, resort to simulation)	
$M/D/1$	$\rho'^2/2(1 - \rho')$	$\rho'/2\mu'(1 - \rho')$
$M/M/1$	$\rho'^2/(1 - \rho')$	$\rho/\mu(1 - \rho')$

Notice the average queue length and queuing delay for an $M/D/1$ queue is half of that for an $M/M/1$ queue, as confirmed by the plot in Figure A3.7. Several other observations are also worthy of note. First, the percentage of idle time = 1 − ρ' = P(system empty) = P_0. For an $M/M/1$ system, P(i units in the system) = $P_0\rho'^i = P_i$. Second, total time in system (W_T) is the combination of queuing delay and service time as mentioned, or $W_q + (1/\mu')$. Third, it is seen that the total number of demands on the system is the combination of those in the queue and those being served = $L_q + \rho'$, and queuing delay is simply $W_q = L_q/\lambda''$. Finally, of significance is that most queuing systems are not subject to closed-form solutions, as already alluded to in the case of $D/M/1$ queue, not to say more complicated ones.

Example
Suppose demand arrives at a rate of $\lambda'' = 0.75$ veh/unit-time and $\rho' = 1$ veh/unit-time, setting aside economic and other considerations, is it more desirable to have constant service time or random service time?

According to the formulas given above, we construct this tabular calculation:

queue	W_q
$M/D/1$	$\frac{0.75}{2(1)(1-0.75)} = 1.5$
$M/M/1$	$\frac{0.75}{1(1-0.75)} = 3$

Thus the obvious answer is to go for constant service time since it results in half of the wait time (and queue length) alluded to earlier. The astute reader would have arrived at this conclusion directly from the queuing formula table above, without substituting any numbers. ■

An example of a multi-server system is the random-arrival, random-service $M/M/p$ queuing model, where the percentage of idle time is

$$P_0 = \frac{1}{\sum_{i=0}^{p-1} \frac{(\lambda''/\mu')^t}{i!} + \frac{(\lambda''/\mu')^p}{p!} \frac{1}{1 - \lambda''/\mu' p}} \tag{A3.7}$$

$$P_i = \begin{cases} [(\lambda''/\mu')_i/i!]P_0 & \textit{if } 0 \le i \le p \\ [(\lambda''/\mu')^i_0/p!p^{i-p}]P & \textit{if } i \ge p \end{cases}$$

$$L_q = \frac{(\lambda''/\mu')^i(\lambda''/\mu' p)P_0}{p!(1 - \lambda''/\mu' p)^2} \tag{A3.8}$$

Example
Given $\lambda'' = 2$ veh/Min, and $\mu' = 3$ veh/Min and $p = 2$, what is the percentage time the system is empty? How about with one vehicle in the system and with two vehicles in the system?

$$P_0 = \frac{1}{\sum_{i=0}^{1} \frac{(2/3)^i}{i!} + \frac{(2/3)^2}{2!} \frac{1}{1 - (2/(2)(3))}} = 0.5 \tag{A3.9}$$

$P_1 = [(2/3)^1/1!](0.5) = 0.333$ and $P_2 = [(2/3)^2/2!2^{2-2}](0.5) = 0.111$. ■

In many queuing systems, an arrival who finds all servers occupied is, for all practical purposes, lost to the system. For example, suppose someone calls in a fire alarm and no fire engines are available; an engine has to be called in from a neighboring town or the fire will simply burn out of control. The result is that the demand evaporates from the local fire station. Thus, a request for a fire engine that occurs when no engines are available may be considered lost to the system. If demand arrivals who find all servers occupied leave the system, we call the system blocked-demands cleared. Assuming that interarrival times are exponential, such a system may be modeled as an $M/G/p/p$ system, where G stands for general distribution (which includes the above cases of random and constant service) and all p servers are identically distributed. The extra p at the end of the notation stands for a capacity of serving up to p demands only.

For an $M/G/p/p$ system, L', W_T, L_q and W_q are of limited interest. Since a queue can never occur, hence $L_q = W_q = 0$. We let $1/\rho'$ be the mean service time and λ'' be the arrival rate. Then $W_T = 1/\mu'$. In most blocked-demands cleared systems, primary interest is focused on the fraction of all demand arrivals who are turned away. Hence an average of $\lambda''P(p)$ arrivals per unit time will be lost to the system, where $P(p)$ is generally referred to as the loss probability. Since an average of λ'' $(1 - P(p))$ arrivals per unit time will actually enter the system, we may conclude that the average queue length in the system (including the ones being served) is

$$L' = \frac{\lambda''(1 - P(p))}{\mu'}$$

For an $M/G/p/p$ system, it can be shown that $P(p)$, the percentage time p-servers are occupied, depends on the service time distribution only through its mean $1/\mu'$. This fact is known as Erlang's loss formula. In other words, any $M/G/p/p$ system with an arrival rate λ'' and service time of $1/\mu'$ will have the same value of $P(p)$ (Winston 1994) and

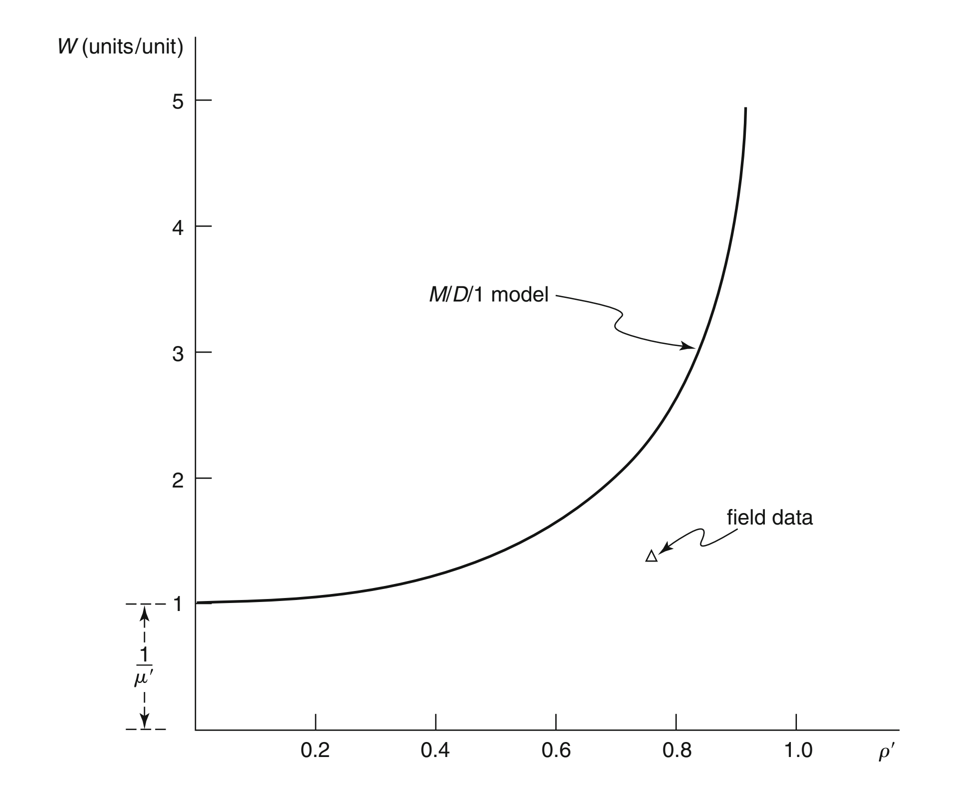
$$P(p) = (\rho'^p/p!)\Bigg/\sum_{k=1}^{p} \rho'^k/k!$$

The Erlang loss formula, including the loss probability $P(p)$, has been computed in terms of nomographs for everyday use (Cooper 1980). The Erlang loss formula is important in locating such facilities as fire stations, as shown in the "Stochastic Facility Location" subsection of the "Measuring Spatial Separation" chapter in Chan (2005).

C. Choosing a Queuing Discipline

With so many queuing disciplines, the logical question at this juncture is "Which model best describes our situation?" Obviously, the answer is not simple, since this is where theory meets application. Perhaps the best way to answer this question is to return to the field data we collected in Table A3.1 for the air terminal. In this data set, we assume that only one ground crew is available, who takes exactly a half hour to service an aircraft. Notice that only the first two columns need to be compiled, the rest can be calculated. Take the first row for example. Since it is the first aircraft that arrived, no wait is necessary for it to be serviced. The total time in the system consists of only the service time, which is a half hour. The only demand traffic is this first aircraft, which constitutes the total number of vehicles in the system. Since this is the only aircraft thus far, there is no interarrival time yet. Following this line of logic, the reader is invited to go through another few lines and derive the rest of the columns from the first two in Table A3.1. From the discussions in Section II, we were inconclusive about whether demand arrivals are random. The service pattern, however, appears to be deterministic, since it takes precisely one half hour to service an aircraft. Remember that the average arrival rate was 0.75 vehicles per unit-time. We compute the rest of the parameters below: service rate = $\mu' = 1$ veh/unit, utilization factor = $\rho' = 0.75$, total time in system = 16.9/12 = 1.41 units, and (interarrival time = 15/11 = 1.26). Based on these calculations, both the theoretical curve and empirical data point can be plotted. We now overlay the experimental data point on the $M/D/1$ theoretical plot of W_T against ρ' in Figure A3.9. Now the question that arises is: How good is the $M/D/1$ model in predicting field data? The answer is again inconclusive, since only one data point is available. Additional information is required for a more definitive answer to the question. This example, simple as it may be, shows that choosing a queuing discipline to fit the data—one of the most important tasks—is by no means straightforward.

Figure A3.9 FIELD DATA ON $M/D/1$ CURVE



One obvious solution is to collect more data. This will allow more validation points to be plotted, not just the one shown in Figure A3.9. This does not, however, address the validation question any better than what has been illustrated. Use of statistical tests, such as the chi-square goodness-of-fit statistic, would help. But again the answer ultimately rests with the judgment of the model builder. Even if the data pass the statistical test, there remains the following question: "Are the data collected in isolation or are they part of a larger queuing system (such as a system preceded by aircraft landing patterns)?" If it is part of a larger system, the input rate to the gate cannot be random in spite of the statistical tests. The input stream in this case is likely to be influenced by the landing policy, or technically speaking, conditioned upon the landing pattern. This opens the question of whether two queues—aircraft landing and service at the gate—are actually in tandem. It is easy to see, again through this simple example, that choosing the appropriate queuing discipline remains an art (not a science.)

V. MARKOVIAN PROPERTIES

Regarding the state transition equations shown above in Sections I and III, both examples are special cases of the continuous time Markov process, where the following Markovian properties are discerned:

1. The conditional probability of any future state $X(t_{k+1}) = j$, given any past state $X(t_0) = i_0, \ldots, X(t_{k-1}) = i_{k-1}$ and the present state $X(t_k) = i_k$ is independent of past states and depends only on the present state of the process:

$$P[X(t_{k+1}) = j \mid X(t_k) = i_k, X(t_{k-1}) = i_{k-1}, \ldots, X(t_0) = i_0] = P[X(t_{k+1}) = j \mid X(t_k) = i_k]$$

This is usually referred to as the memoryless property.

2. The process is stationary if the transitional probabilities above depend only on the time interval between the events rather than on absolute time t: $P[X(t_2) = j \mid X(t_1) = i] = P[X(t_2 - t_1) = j \mid X(0) = i]$, for all i, j, t_1, t_2 ($t_1 < t_2$). In other words, the starting time of the process is unimportant in comparison to the amount of time that has elapsed $t = t_2 - t_1$. Given these two properties, a Markov process is completely described by its transition probabilities $p_{ij}(t) = P[X(t) = j \mid X(0) = i]$. Notice the concept of stationarity is similar to that in time series (see the "Spatial Time Series" chapter in Chan [2005]).

Examples

Consider a Poisson process in which $X(0) = 0$ gives rise to a distribution $p_{0x}(t) = (\lambda''t)^x e^{-\lambda''t}/x!$, for $x = 0, 1, 2, 3, \ldots, n$. Suppose for $X(t_1) = 2$, $p_{2X'}(\tau) = (\lambda''\tau)^{X'} e^{-\lambda''\tau}/X'!$, where $\tau = t - t_1$ and $X' = x - 2$. For the time interval t or τ, both are the same distribution function in spite of different start times (0 vs. t_1) and different initial conditions

($X(0) = 0$ vs. $X(t_1) = 2$). Having two vehicle arrivals already merely shifted the cumulative curve up by 2 for the initial state, irrespective of past history of the arrival pattern. With a transformation of state variables, $X^i(t_1) = 0$, which is the same as $x(0) = 0$. In diagrammatic form, the same cumulative and state transition diagram for Poisson process can be overlaid on top of Figure A3.6: one starts at time 0 and the other at t_1. This example illustrates the memoryless and stationary properties of a Markovian system. Also for the same example, $p_{00}(\tau) = p_{22}(\tau) = e^{-\lambda''\tau}$ for $0 \leq \tau \leq \infty$; or the times between events in a Poisson process are all negative exponentially distributed with the same parameter λ''. This is irrespective of whether we start with $t = 0$ (when the system is idle) or $t = t_1$ (after two arrivals have been logged).

Similarly, for $M/M/1$ queue, the initial state can be $X(0) = 0$ or $X(t_1) = 2$, and the identical distribution $P_i = \rho'^i(1 - \rho')$ results in the steady state. Graphically speaking, cumulative and state transition diagrams for both $M/M/1$ queues can again be overlaid on top of Figure A3.6 to illustrate the memoryless and stationary properties. In this Figure, both cumulative/departure and state transition diagrams look the same to the right of the starting point. The only difference is that two arrivals and one departure have occurred in the latter case. Again, the history of the process is adequately represented by the initial state. $X(0)$ and $X(t_1)$ and does not depend on the history prior to $t = 0$ and $t = t_1$. ■

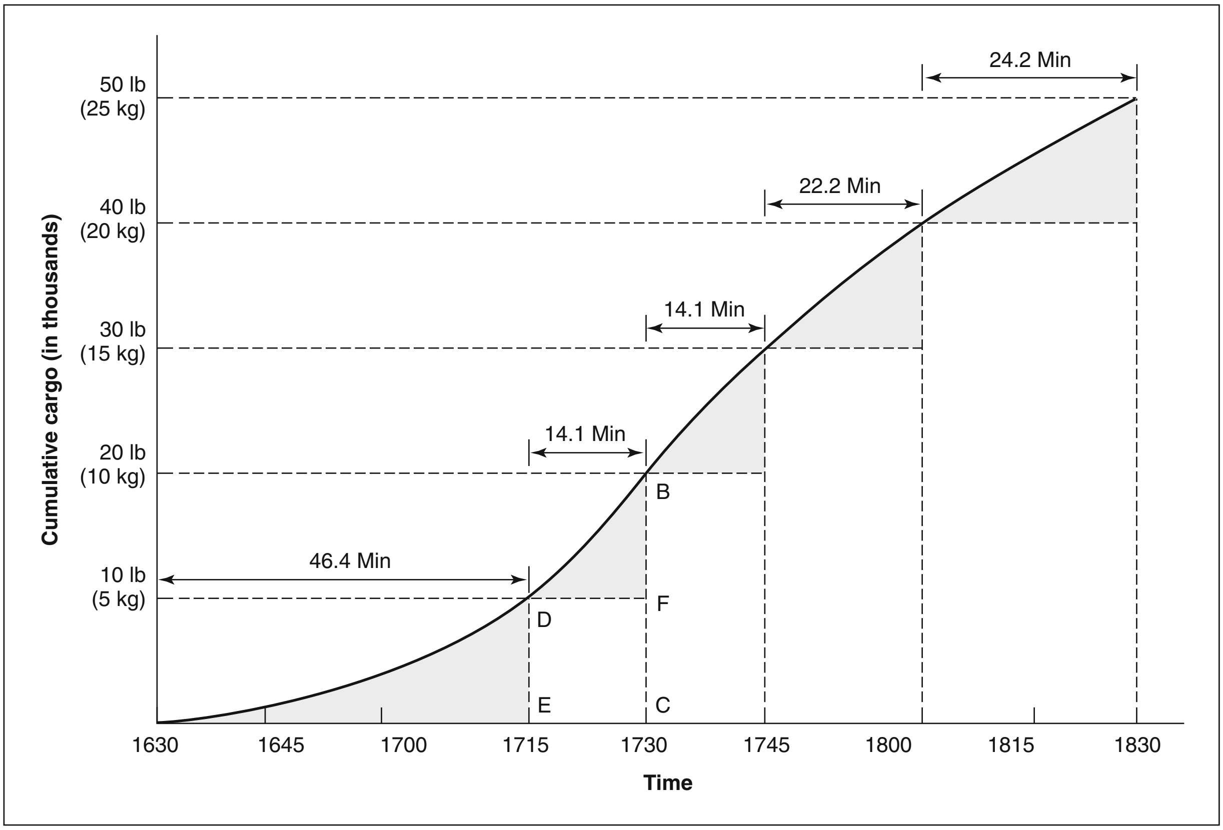
VI. MARKOVIAN PROPERTIES OF DYNAMIC PROGRAMMING

Perhaps we can illustrate Markovian property even better with regular dynamic programming (DP), which is a set of deterministic state transition equations in which a decision variable is built into a Markovian system for optimization purposes. The best way to describe DP is through examples.

A. Vehicle Dispatching Example

We are to construct a timetable for dispatching a cargo aircraft toward the end of a business day. The cargo carrying capacity of the vehicle is 30,000 lbs (15,000 kg). Due to space limitations, we do not allow more than a vehicle load of cargo (30,000 lbs) to accumulate at the loading dock. Our objective is to minimize the cost-of-operation and delay cost experienced by the cargo consignee (who either receive the cargo early or late). One can think of constructing a timetable as making a series of decisions as to whether or not to dispatch at each instance when 10,000 lb (5,000 kg) of cargo are accumulated. Figure A3.10 shows a demand-arrival pattern from 4:30 pm to 6:30 pm. We define stages k as the times at which dispatching decisions are made; and states X_k as the accumulated inventory of cargo at the loading dock when a dispatching decision is reviewed. The rules of engagement are that when the vehicle is filled up, it has to be dispatched. At the end of the business day, all cargo has to be dispatched in order to clear the dock. The decision variable y_k is a binary 0-1 variable: 1 stands for "dispatch" and 0 for "hold". The operating cost function $c(y_k)$ can be represented by Table A3.4. Thus if the decision is to dispatch, it will invariably cost 6,000 dollars to fly the aircraft. On the other hand, the delay cost is varying depending on the shape of the demand arrival curve shown in Figure A3.10. These delay costs can be calculated below.

Figure A3.10 DEMAND ARRIVAL PATTERN



1. Markovian Properties. Delay to consignee can be represented in pound-minutes (kg-Min), graphically depicted as the shaded wedges in Figure A3.10 if the decision is always to dispatch at each decision point. Assume a delay wedge, measured in lb-Min (kg-Min), can be approximated by a triangle. When the decision is no at the first decision point *E*, the delay cost will be the bigger triangle *ABC* rather than the two smaller triangles *ADE* and *DBF*. Would this refer us two stages back rather than just one stage, thus destroying the Markovian property? Graphical examination seems to confirm this, since between the two the rectangle *DFCE* is missing. But a little transformation of the cost accounting procedure will guarantee the Markovian property. To show this, a little calculation will help:

Table A3.4 OPERATING AND DELAY COSTS FOR DISPATCHING EXAMPLE

		Incremental cost ($) of operation $c(y_k)$ for decision point (stage) k					
		0	**1**	**2**	**3**	**4**	**5**
$y_k =$	1 dispatch	—	6000	6000	6000	6000	6000
	0 hold	4640	1410	1410	2220	2420	—

Average delay-cost for a vehicle load of cargo in dollars
= (avg delay to a lb [kg]) (value of time for a veh load of cargo)
= (abscissa/2) (time value for an avg veh load of 10,000 lb [5,000 kg])
↑ ↑
in Min in $/Min
= (abscissa/2) ($200/Min)

Average delay for triangle *ABC*
= *AC*/2
= *AE*/2 + *EC*/2
= (avg delay of triangle *ADE*) + (avg delay of triangle *DBF*).

This transformation effectively measures delay in terms of time on the horizontal axis. By accounting for an *average* load of cargo, the missing rectangle dilemma disappears. Each decision y_k over time interval $(t_k - t_{k-1})$ is now separable, and costs are cumulative (incremental from the last "running sum.") Notice the assumption on the size of an average cargo load is unimportant, as long as we use the same vehicle load consistently. For an average load of 10,000 lb., the delay costs are shown in Table A3.4.

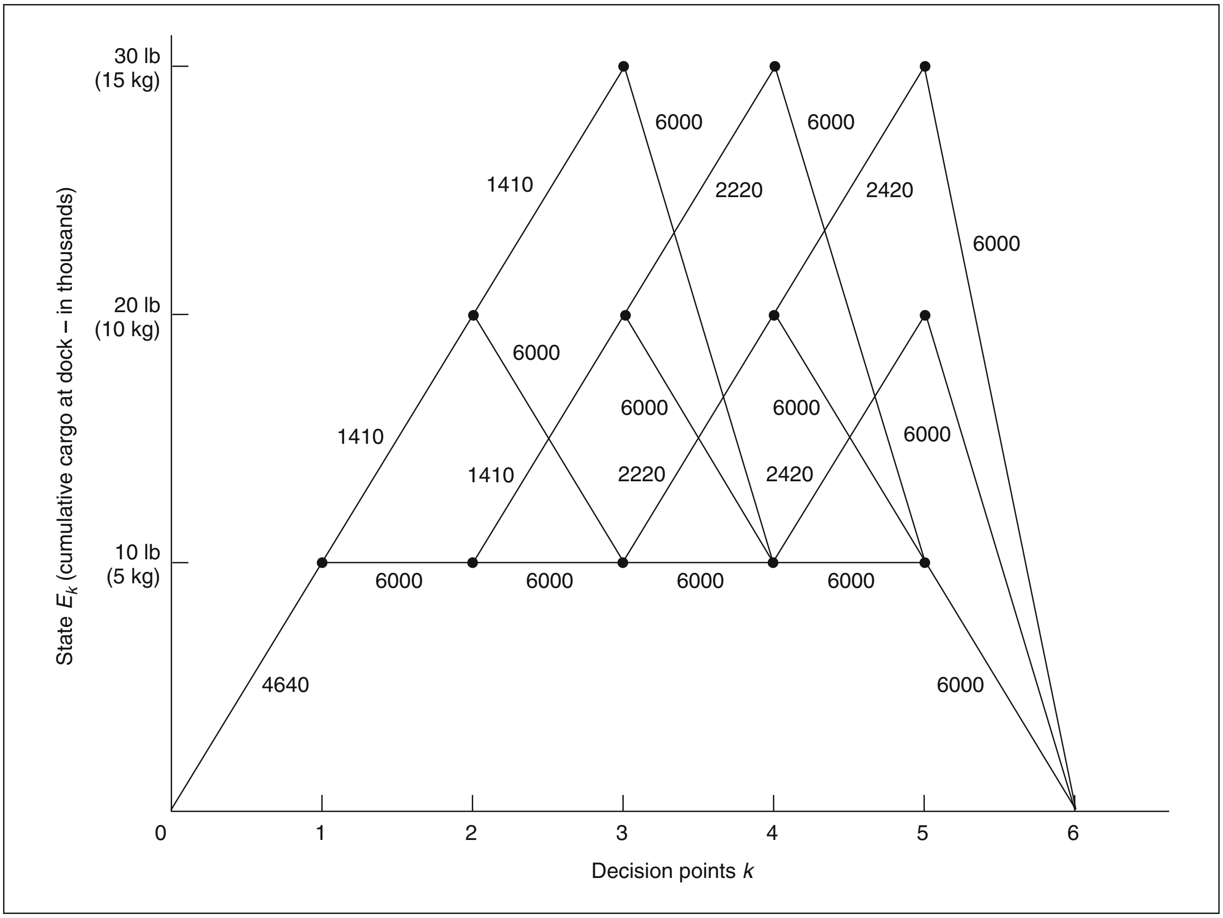
Here in this example, all arc costs are anticipatory and discretionary. However, there is an unavoidable overhead cost of $(4640 + 1410 + 1410 + 2220 + 420) = $12,000, in other words the sum of all shaded triangles in Figure A3.10. They are the result of our policy that a dispatch decision will not be reviewed until 10,000 pounds of cargo have arrived. Notice the $12,100 is implicit and does not need to appear in our objective function. Instead of measuring anticipatory delay by a triangular wedge, however, it can be shown that a retrospective delay can be measured by such rectangles as that formed by the points *DFCE*. In this case, only historic cost is accounted for and the anticipatory overhead-cost evaporates. The return function is now both a function of the decision variable x_k and the state variable X_k: $c(x_k, X_k)$. The Markovian property is automatically upheld, and hence no cost transformation is necessary.

2. Solution Algorithm. Now that we have established a Markovian system, the problem is ready for solution. To properly solve this problem, however, a state-stage diagram needs to be constructed. This diagram—pictured in Figure A3.11—is derived from the demand arrival curve, but the linkage stops at that point. Similar to the Markovian state-transition diagram depicted in Figure A3.1 and Figure A3.6, the state stage diagram can be described by a set of state transition equations. The steady-state equations (for measuring cargo in pounds) look like

$$X_k = X_{k-1} + 10000(1 - y_{k-1}[X_{k-1}/10000]) \qquad \text{(A3.10)}$$

where X_k is the state variable (in lbs of cumulative cargo at the dock) at decision point k. These equations are subject to boundary conditions at $k = 0$ and 5: when $k = 0$, $y_k = 0$; and at $k = 5$, $y_5 = 1$. Numerical examples of these steady-state transition equations can be provided. For a hold decision at decision point 0, the accumulated cargo at the next decision point will be 10000 lbs. as confirmed by $X_1 = X_0 + 10000(1 - y_0[X_0/10000]) = 0 + 10000(1 - 0[0/10000]) = 10000$. Another hold decision at $k = 1$ will result in 20,000 lb cargo at the dock by decision point $k = 3$:

Figure A3.11 STATE STAGE DIAGRAM



$X_2 = X_1 + 10000(1 - \ y_1[X_1/10000]) = 10000 + 10000(1 - 0\,[10000/\ 10000]) = 20000$, and so forth.

Solution of the system of equations can be carried out working backwards:

$$f_5^*(X_5 = 10000) = 6000, \text{ where } y_5 = 1$$
$$f_5^*(X_5 = 20000) = 6000, \text{ where } y_5 = 1$$
$$f_5^*(X_5 = 30000) = 6000, \text{ where } y_5 = 1$$

where f_k is the running sum or the criterion function. In tabular form

$k = 6$	X_5	$f_5^*(X_5)$	y_5
	10000	6000	1
	20000	6000	1
	30000	6000	1

$$\begin{aligned}
f_4^*(10000) &= [c(y_4) + f_5^*(X_5)] \\
&= \text{Min}\{[6000 + f_5^*(10000)], [2420 + f_5^*(20000)]\} \\
&= \text{Min }[(6000 + 6000), (2420 + 6000)] = 8420, \text{ where } y_4 = 0
\end{aligned}$$

$$\begin{aligned}
f_4^*(20000) &= \text{Min }\{[6000 + f_5^*(10000)], [2420 + f_5^*(30000)]\} \\
&= \text{Min }(6000 + 6000, 2420 + 6000) = 8420, \text{ where } y_4 = 0
\end{aligned}$$

$$f_4^*(30000) = \text{Min }[6000 + f_5^*(10000)] = 12000, \text{ where } y_4 = 1$$

The recursive function for these three equations is the common form

$$f_4^*(X_4) = \underset{y_4=1,\,0}{\text{Min}}\,[f_4(X_4, y_4)] = \underset{y_4=1,\,0}{\text{Min}}\,[c(y_4) + f_5^*(X_5)]$$

In tabular form:

$k = 5$	X_4	$f_4{}^*(X_4)$	y_4
	10000	8420	0
	20000	8420	0
	30000	12000	1

The remaining iterations can be tabulated as follows:

	X_{k-1}	$f_{k-1}^{\ *}(X_{k-1})$	$y_{k=1}$
	10000	10640	0
$k = 4$	20000	14220	0
	30000	14420	1
	10000	15630	0
$k = 3$	20000	15830	0
$k = 2$	10000	17240	0
$k = 1$	0	21880	0

In general, the recursive equation is $f_{k-1}^{\ *}(X_{k-1}) = \underset{y_{k-1}=1,\,0}{\text{Min}}\,[c(y_{k-1}) + f_k^*(X_k)]$ for $k = 5, 4, \ldots, 1$. We trace back using the state transition Equation A3.10, with the assistance of the tabular computations for $k = 5, 4, 3, 2, 1$ above. Starting with the boundary condition at $k = 1$: $X_0 = 0$, $y_0 = 0 \rightarrow X_1 = 10000$, $y_1 = 0 \rightarrow X_2 = 20000$, $y_2 = 0 \rightarrow X_3 = 30000$, $y_3 = 1 \rightarrow X_4 = 10000$, $y_4 = 0 \rightarrow X_5 = 20000$, $y_5 = 1$. Thus the dispatch timetable is to send an aircraft out at decision points 3 and 5. This translates to about 5:40 p.m. and 6:30 p.m. While the example has been worked out using backward recursion, it can be shown that it can be solved equally well using forward recursion.

B. Principle of Optimality

In summary, a Markovian system such as the example above can be optimized using the principle of optimality, which we shall recap simply: After k decisions have been made, the effect of the remaining $n - k$ stages on the total criterion-function f depends only on the state at stage k and the final $n - k$ decisions. In other words, an optimal policy for the remaining stages is independent of the policy adopted in previous stages. This principle is the basis of many solutions to an optimal control problem such as the vehicle dispatching one under discussion.

The general forward recursion equations of DP can now be expressed as follows (backward recursion is similar):

$$\begin{aligned} &f_k^*(X_k) = \text{opt}_x\,[r_k(x_k) \circ f_{k-1}^*(X_{k-1})] \\ &\text{s.t. } X_{k-1} = h'(x_k, X_k) \quad k = 1, 2, \ldots, n \end{aligned}$$

where *opt* is a subproblem of maximization or minimization, $\circ$ stands for addition or multiplication, X is the state variable, k is the stage variable, x is the decision variable, f is the criterion function, $r(x_k)$ is the return function[2], and h' is the state transition function. In the vehicle dispatching example, these terms can be illustrated below. Notice that each minimization subproblem can be solved by simple arithmetic calculations:

Formal terms	Illustration in the vehicle-dispatching example
$X_{k-1} = h'(X_k, x_k)$	$X_k = X_{k-1} + 10000\ (1 - x_{k-1}[X_{k-1}/10000])$
$r_k(x_k)$	$c(y_k)$
f_k	$c(y_k) + f_{k+1}$
subproblem solution	arithmetics

Based on this example, these general observations can be made regarding the optimization of a Markovian system. First the system has to be decomposable:

$$f_n(x_n, X_n) = f_0(x_0, X_0) \circ r(x_1, X_1) \circ r(x_2, X_2) \circ \ldots \circ r(x_n, X_n)$$

where the return function r is separable or $r(X, Y) = r_1(X, r_2(Y))$, in other words, return at each stage is independent of previous decisions and subsequent decisions, and r_1 is monotonically nondecreasing (or nonincreasing) relative to its second argument. Then local decision at each stage "adds up" to an overall multistage decision. An example of a separable function is $(x_1 + x_2^3)$, or x_1x_2. As an opposite example $(x_1 \ln x_2 + x_2)$ illustrates a non-decomposable function.

Second, the system has to be Markovian:

$$\begin{aligned} f_n^*(x_n, X_n) &= \underset{x_1,\ldots,x_n}{opt}\ \{f_n(x_n, X_n)\} = \underset{x_n}{opt}\ \{\ \underset{x_1,\ldots,x_{n-1}}{opt}\ [f_{n-1}(x_{n-1}, S_{n-1})]\} \\ &= \underset{x_n}{opt}\ \langle\ \underset{x_{n-1}}{opt}\ \{\ \underset{x_1,\ldots,x_{n-2}}{opt}\ [f_{n-2}(x_{n-2}, S_{n-2})]\}\ \rangle \ldots \text{etc.} \end{aligned}$$

This says that after k decisions have been made, the effect of the remaining $n - k$ stages on the total criterion function depends only on the state X_k, and the final $n - k$ decisions $x_{k+1}, \ldots, x_{n-1}, x_n$. Put it in a different way, an optimal policy for the remaining stages is independent of the policy adopted in previous stages.

These two properties—separability and memorylessness—are prerequisites to the optimization of a Markovian system. We have described them above in terms of forward recursion. Similar arguments can be made for backward recursion as well.[3] Even though a deterministic system was used in the DP example, we will show below how this can be generalized to a probabilistic system.

VII. MARKOVIAN DECISION PROCESSES

Infinite-horizon probabilistic dynamic-programming problems are called **Markovian decision processes** (MDP) (Winston 1994). An MDP is described by four types of information: state space, decision set, transition probabilities, and expected rewards. At the beginning of each period, the MDP is in some state i, where i is a member of the state space $X' = \{1, 2, \ldots, n\}$. For each state i, there is a finite set of allowable decisions, $D(i)$. Suppose a period begins in state i, and a decision $d'' \in D(i)$ is chosen. Then with probability $\pi(j \mid i, d'')$, the next period's state will be j. The next period's state depends only on the current period's state and on the decision chosen during the current period (and not on previous states and decisions). During a period in which the state is i and a decision $d'' \in D(i)$ is chosen, an expected reward of $r(i, d'')$ is received.

A. Policy Iteration

In an MDP, what criterion should be used to determine the correct decision? Answering this question requires that we discuss the idea of an optimal policy for an MDP. A policy is a rule that specifies how each period's decision is chosen. A policy $\tilde{\delta}$ is a stationary policy if whenever the state is i, the policy $\tilde{\delta}$ chooses (independently of the period) the same decision (call this decision $\delta(i)$). If a policy δ^* has the property that for all $i \in X'$, the optimal expected value of the decision at state i, $Z(i)$, is the same as that obtained from the policy δ^*, $Z_{\delta^*}(i)$, or $Z(i) = Z_{\delta^*}(i)$, then δ^* is an optimal policy.

1. Value Determination. Let us determine a system of linear equations that can be used to find $Z_\delta(i)$ for $i \in X'$ for any stationary policy δ. If $\delta(i)$ stands for the decision chosen by the stationary policy δ whenever the process begins a period in state i, then $Z_\delta(i)$ can be found by solving the following system of n linear equations, the value determination equations:

$$Z_\delta(i) = r(i, \delta(i)) + \ell' \sum_{j=1}^{n} \pi(j \mid i, \delta(i)) Z_\delta(j) \quad i = 1, \ldots, n \qquad \text{(A3.11)}$$

Here we are discounting rewards by assuming that a dollar reward received during the next period will have the same value as a reward of ℓ' dollars $(0 < \ell' < 1)$ received during the current period. This is equivalent to assuming that the decision maker wishes to maximize expected discounted reward. Then the expected discounted reward earned during an infinite number of periods consists of $r(i, \delta(i))$ (the expected reward earned during the current period) plus ℓ' times the expected discounted reward. This discounted value includes the reward to the beginning of the next period, earned from the next period onward. But with probability $\pi(j\,|\,i, \delta(i))$, we will begin the next period in state j and earn an expected discounted reward, back to this next period, of $Z_\delta(j)$. Thus the expected discounted reward, discounted back to the beginning of the next period and earned from the beginning of the next period onward, is given by $\sum_{j=1}^{n} \pi(j\,|\,i, \delta(i))Z_\delta(j)$. Equation A3.11 now follows. Notice its similarity to the recursion equation in deterministic dynamic programming.

Example
To illustrate the use of the value determination equations, let us consider a site relocation example characterized by Table A3.5, which shows the probable degradation of a site over time as demand patterns change. We consider the following stationary policy for the site relocation example: $\delta(E) = \delta(G) = N$ and $\delta(A) = \delta(B) = Y$, where Y stands for relocation and N stands for no relocation. This policy relocates a bad (B) or average (A) site to an excellent (E) site at a cost (negative reward) and does not relocate a good (G) or excellent (E) site. For this policy and the given rewards and discount factor, Equation A3.11 yields the following four equations:

$$
\begin{array}{ll}
Z_\delta(E) = 100 + 0.9(0.7\,Z_\delta(E) + 0.3\,Z_\delta(\mathrm{G})) & Z_\delta(A) = 100 + 0.9(0.7\,Z_\delta(E) + 0.3\,Z_\delta(\mathrm{G})) \\
Z_\delta(G) = 80 + 0.9(0.7\,Z_\delta(G) + 0.3\,Z_\delta(A)) & Z_\delta(B) = -100 + 0.9(0.7\,Z_\delta(E) + 0.3\,Z_\delta(G))
\end{array}
$$

The last two equations suggest that with probabilities 0.7 and 0.3, the excellent site will remain "excellent" or become "good" respectively at the beginning of the next period. Solving these equations yields $Z_\delta(E) = 687.81$, $Z_\delta(G) = 572.19$, $Z_\delta(A) = 487.81$, and $Z_\delta(B) = 487.81$. In other words, following the stationary policy as outlined above, the expected value of having an excellent, good, average, and bad site can be uniquely determined. ■

2. Howard's Method for Optimal Policy. We now describe Howard's (1960) policy iteration method for finding an optimal stationary policy for an MDP.

Table A3.5 TRANSITION MATRIX OF SITE-RELOCATION EXAMPLE

Present state	Probability that site begins next year as			
of site	Excellent (E)	Good (G)	Average (A)	Bad (B)
Excellent (E)	0.7	0.3		
Good (G)		0.7	0.3	
Average (A)			0.6	0.4
Bad (B)				1.0[a]

[a] A "bad" site remains "bad" until relocation takes place.

Step 1. Policy evaluation: Choose a stationary policy $\tilde{\delta}$ and use the value determination equations to find $Z_\delta(i)$, $i = 1, \ldots, n$.

Step 2. Policy improvement: For all states $i = 1, \ldots, n$, compute

$$Z_\delta' = \operatorname*{Max}_{d''\in D(i)} (r(i, d'') + \ell' \sum_{j=1}^{n} \pi(j \mid i, d'')Z_\delta(j)) \tag{A3.12}$$

Since we can choose $d'' = \delta(i)$ for $i = 1, \ldots, n$, $Z_\delta'(i) \geq Z_\delta(i)$. If $Z_\delta'(i) = Z_\delta(i)$ for $i = 1, \ldots, n$, then $\tilde{\delta}$ is an optimal policy. If $Z_\delta'(i) > Z_\delta(i)$ for at least one state, $\tilde{\delta}$ is not an optimal policy. In this case, modify $\tilde{\delta}$ so that the decision in each state is the decision attaining the maximum in Equation A3.12 for $Z_\delta'(i)$. This yields a new stationary policy δ' from which $Z_\delta'(i) \geq Z_\delta(i)$ for $i = 1, \ldots, n$, and for at least one state i', $Z_{\delta'}(i') > Z_\delta(i')$. Return to Step 1, with policy $\tilde{\delta}'$ replacing policy $\tilde{\delta}$. The policy iteration method is guaranteed to find an optimal policy after evaluating a finite number of policies. We now use the policy iteration method to find an optimal stationary policy for the site relocation example.

Example

We begin with the stationary policy as mentioned: $\delta(E) = \delta(G) = N$ and $\delta(A) = \delta(B) = Y$. For this policy, we have already found that $Z_\delta(E) = 687.81$, $Z_\delta(G) = 572.19$, $Z_\delta(A) = 487.81$, and $Z_\delta(B) = 487.81$. We now compute $Z_\delta'(E)$, $Z_\delta'(G)$, $Z_\delta'(A)$, and $Z_\delta'(B)$. Since N is the only possible decision in E according to Table A3.5, $Z_\delta'(E) = Z_\delta(E) = 687.81$ and $Z_\delta'(E)$ is attained by the decision N. State G can become state E with a relocation (Y) or stay at G with no relocation (N):

$$Z_\delta'(G) = \text{Max}\begin{cases} -100 + 0.9(0.7\,Z_\delta(E) + 0.3\,Z_\delta(G)) = 487.81 & (Y) \\ 80 + 0.9(0.7\,Z_\delta(G) + 0.3\,Z_\delta(A)) = Z_\delta(G) = 572.19^* & (N) \end{cases} \tag{A3.13}$$

Thus, $Z_\delta'(G) = 572.19$ is attained by the decision N which incurs a larger reward. State A can become E or remain at A involving a Y or N decision respectively:

$$Z_\delta'(A) = \text{Max}\begin{cases} -100 + 0.9(0.7\,Z_\delta(E) + 0.3\,Z_\delta(G) = 487.81 & (Y) \\ 50 + 0.9(0.6\,Z_\delta(A) + 0.4\,Z_\delta(B)) = Z_\delta(A) = 489.03^* & (N) \end{cases} \tag{A3.14}$$

Thus $Z_\delta'(A) = 489.03$ is attained by the decision N. B can be upgraded to E or remain at B:

$$Z_\delta'(B) = \text{Max}\begin{cases} -100 + 0.9(0.7\,Z_\delta(E) + 0.3\,Z_\delta(G)) = 487.81^* & (Y) \\ 10 + 0.9\,Z_\delta(B) = 449.03 & (N) \end{cases} \tag{A3.15}$$

Thus $Z_\delta'(B) = Z_\delta(B) = 487.81$.

We have found that $Z_\delta'(E) = Z_\delta(E)$, $Z_\delta'(G) = Z_\delta(G)$, $Z_\delta'(B) = Z_\delta(B)$, and $Z_\delta'(A) > Z_\delta(A)$. The policy $\tilde{\delta}$ is not optimal, and the policy $\tilde{\delta}'$ given by $\delta'(E) = \delta'(G) = \delta'(A) = N$, $\delta'(B) = Y$, is an improvement over δ. Notice the new policy relocates only when the site is bad. We now return to Step 1 and solve the value determination equations for δ'. From Equation (A3.11), the value determination equations for δ' are

$$Z_{\delta'}(E) = 100 + 0.9(0.7\,Z_{\delta'}(E) + 0.3\,Z_{\delta'}(G)) \qquad Z_{\delta'}(A) = 50 + 0.9(0.6\,Z_{\delta'}(A) + 0.4\,Z_{\delta'}(B))$$
$$Z_{\delta'}(G) = 80 + 0.9(0.7\,Z_{\delta'}(G) + 0.3\,Z_{\delta'}(A)) \quad Z_{\delta'}(B) = -100 + 0.9(0.7\,Z_{\delta'}(E) + 0.3\,Z_{\delta'}(G))$$

Solving these equations, we obtain $Z_{\delta'}(E) = 690.23$, $Z_{\delta'}(G) = 575.50$, $Z_{\delta'}(\mathrm{A}) = 492.35$, and $Z_{\delta'}(\mathrm{B}) = 490.23$. Observe that in each state i, $Z_{\delta'}(i) > Z_{\delta}(i)$. We now apply the policy iteration procedure to δ'. We compute $Z'_{\delta'}(E) = Z_{\delta'}(E) = 690.23$, N being the only decision.

$$Z'_{\delta'}(G) = \text{Max}\begin{cases} -100 + 0.9(0.7Z_{\delta'}(E) + 0.3Z_{\delta'}(G)) = 490.23 & (Y) \\ 80 + 0.9(0.7Z_{\delta'}(G) + 0.3Z_{\delta'}(A)) = Z_{\delta'}(G) = 575.50^* & (N) \end{cases} \tag{A3.16}$$

for transitions to E and G respectively. Thus, $Z'_{\delta'}(G) = Z_{\delta'}(G) = 575.50$ is attained by the decision N.

$$Z'_{\delta}(A) = \text{Max}\begin{cases} -100 + 0.9(0.7Z_{\delta'}(E) + 0.3Z_{\delta'}(G)) = 490.23 & (Y) \\ 50 + 0.9(0.6Z_{\delta'}(A) + 0.4Z_{\delta'}(B)) = Z_{\delta'}(A) = 492.35^* & (N) \end{cases} \tag{A3.17}$$

for transitions to E and A. Thus $Z_{\delta}'(\mathrm{A}) = Z_{\delta'}(\mathrm{A}) = 492.35$ is attained by the decision N.

$$Z_{\delta'}(B) = \text{Max}\begin{cases} -100 + 0.9(0.7Z_{\delta'}(E) + 0.3Z_{\delta'}(G)) = 490.23^* & (Y) \\ 10 + 0.9Z_{\delta'}(B) = 451.21 & (N) \end{cases} \tag{A3.18}$$

for states E and B. Thus $Z'_{\delta'}(B) = Z_{\delta'}(B) = 490.23$ is attained by Y.

For each state i, $Z'_{\delta'}(i) = Z_{\delta'}(i)$. Thus δ' is an optimal stationary policy. In order to maximize expected discounted rewards (profits), a bad site should be relocated, but an excellent, good, or average site should not be relocated. If we began period 1 with an excellent location, an expected discounted reward of \$690.23 dollars could be earned and so on. ■

B. Reward Per Period

Linear programming can be used to find a stationary policy that maximizes the expected per-period rewards earned over an infinite horizon. Consider a decision rule or policy δ that chooses decision $d''\in D(i)$ with probability $P_{id''}$ during a period in which the state is i. A policy δ' will be a stationary policy if each $P_{id''}$ equals 0 or 1. To find a policy that maximizes expected reward per period over an infinite horizon, let $P_{id''}$ be the fraction of all periods in which the state is i and decision $d''\in D(i)$ is chosen. Then the expected reward per period is to be optimized:

$$\text{Max}\sum_{i=1}^{n} \sum_{d''\in D(i)} P_{id''}r(i, d'')$$

What constraints must be satisfied by the $P_{id''}$? First, all $P_{id''}$s must be non-negative, or $P_{id''} \geq 0$ for $i = 1, \ldots, n$ and $d''\in D(i)$. Second, sum of the probabilities must add up to unity:

$$\text{Max}\sum_{i=1}^{n} \sum_{d''\in D(i)} P_{id''} = 1$$

Finally, the fraction of all periods during which a transition occurs out of state *j* must equal the fraction of all periods during which a transition occurs into state *j*. This is identical to the restriction on steady-state probabilities for Markov chains (see Equations A3.1 and A3.4):

$$\sum_{d''\epsilon D(j)} P_{jd''}(1-\pi(j\,|\,i,d'')) = \sum_{d''\epsilon D(i)}\sum_{i\neq j} P_{id''} - \pi(j\,|\,i,d'') \quad j=1,\ldots,n \quad \text{(A3.19)}$$

which, after rearranging, yields

$$\sum_{d''\epsilon D(j)} P_{jd''} = \sum_{d''\epsilon D(i)}\sum_{i=j} P_{id''}\pi(j\,|\,i,d'') \quad j=1,\ldots,n \qquad \text{(A3.20)}$$

It can be shown that this LP has an optimal solution in which for each i, at most one $P_{id''} > 0$. This optimal solution implies that the expected reward per period is minimized by a solution in which each $P_{id''}$ equals 0 or 1. Thus the optimal solution to the LP will occur for a stationary policy. For states having $P_{id''} = 0$, any decision may be chosen without affecting the expected reward per period.

Example
For the relocation example above, the corresponding LP looks like

$$\begin{array}{ll}
\text{Max} & 100P_{EN} + 80P_{GN} + 50P_{AN} + 10P_{BN} - 100(P_{GY} + P_{AY} + P_{BY}) \\
\text{s.t.} & P_{EN} + P_{GN} + P_{AN} + P_{BN} + P_{GY} + P_{AY} + P_{BY} = 1 \\
 & P_{EN} = 0.7(P_{EN} + P_{GY} + P_{AY} + P_{BY}) \\
 & P_{GY} + P_{GN} = 0.3(P_{GY} + P_{AY} + P_{BY} + P_{EN}) + 0.7P_{GN} \\
 & P_{AY} + P_{AN} = 0.3P_{GN} + 0.6P_{AN} \\
 & P_{BY} + P_{BN} = P_{BN} + 0.4P_{AN}
\end{array}$$

with all $P_{id''}$ non-negative. It was found that the optimal objective function is at $60. The only non-zero decision variables are $P_{EN} = 0.35$, $P_{GN} = 0.50$, and $P_{AY} = 0.15$. Thus the system is optimized by not relocating from an excellent or good site, but relocating from an average site. Since we are relocating from an average site, the action chosen during a period in which the site is bad is of no importance. It is not surprising the optimal policy reached here is different from those in response to maximizing expected discounted rewards. ■

Additional locational examples of the Markovian decision process can be found in Chapter 3 under the "Stochastic Process" subsection and in "Measuring Spatial Separation" chapter under the "Approximate versus Exact Measure" subsection in Chan (2005).

VIII. RECURSIVE PROGRAMMING

Related to Howard's policy-iteration method is recursive programming, an analogue spearheaded by economists (Day 1973). We will introduce recursive programming via an example here. More general computational treatment is found in the sub-subsection bearing the same name in the "Location Routing" chapter of Chan (2005). Suppose a firm is planning on a multiyear production of two products. Let $x_1(t)$, $x_2(t)$ be the amounts to be produced on each of the two commodities over year $t = 1, 2$, and so forth. Let $I_1(t)$ and $I_2(t)$ be the profits per unit at the end of year t. Let c_1 and c_2 be the resource requirements of one unit of the two commodities. Let b_1 be the yearly combined production quota of the two commodities, as limited by, labor availability for instance. Finally, let $b_2(t)$ be the production budget available at the beginning of year t. Assuming constant return to scale, or c_1 and c_2 are independent of x_1 and x_2, the decision problem at the beginning of year t can be represented by the linear programming problem:

$$I(t) = \underset{x_1,\, x_2}{\text{Max}}\, [I_1(t)x_1 + I_2(t)x_2] \tag{A3.21}$$

$$\begin{aligned} &\text{s.t.} \\ &\quad x_1 + x_2 \le b_1 \\ &\quad c_1x_1 + c_2x_2 \le b_2(t) \\ &\quad x_1, x_2 \ge 0 \end{aligned} \tag{A3.22}$$

The optimal solution of this LP, $x_1(t)$ and $x_2(t)$, will give the production of each commodity in period t.

The expected marginal net-revenue values of the two factors, labor and capital, are given by the dual variables $u_1(t)$ and $u_2(t)$, obtainable from the dual program:

$$I_D(t) = \underset{u_1,\, u_2}{\text{Min}}\, [b_1u_1 + b_2(t)u_2] \tag{A3.23}$$

$$\begin{aligned} &\text{s.t.} \\ &\quad u_1 + c_1u_2 \ge I_1(t) \\ &\quad u_1 + c_2u_2 \ge I_2(t) \\ &\quad u_1, u_2 \ge 0 \end{aligned} \tag{A3.24}$$

A. Existence of Solutions

Let $p_i(t)$ be the market price at the end of year t for each commodity, then the profit from each commodity unit is the net between revenue and cost:

$$I_i(t) = p_i(t-1) - c_i \qquad i = 1, 2 \tag{A3.25}$$

Working capital now is limited to the sales minus overhead (say at a constant amount of C_0 for each period):

$$b_2(t) = \textstyle\sum_i p_i(t-1)\, x_i(t-1) - C_0 \tag{A3.26}$$

For each year, the existing supply is sold at a uniform price in a perfectly competitive market. The price received for each commodity is a function of the total amount of commodity supplied. When a commodity is sold at a positive market price, its price is determined by a linear demand-function, defined by intercept a and slope b:

$$p_i(t) = \text{Max}\{0, a_i + b_i x_i(t)\} \quad i = 1, 2 \qquad \text{(A3.27)}$$

The system of Equations A3.21–A3.27 is defined as a recursive program. The primal-dual LP problems Equations A3.21–A3.24 describe the optimization component, as driven by profitability. Equations A3.26–A3.27 together with the definition (Equation A3.25) describe the feedback mechanism, in other words how the market dynamics work. It is a closed, discrete time, dynamic system of which one may ask such traditional questions as: Do equilibria exist? Are they stable?

The dual LP at any time t are dual feasible so long as

$$\sum\nolimits_i \left[\text{Max}\left\{0, a_i + b_i x_i(t-1)\right\}\right] x_i(t-1) \geq C_0 \qquad \text{(A3.28)}$$

In other words, if total revenue in the preceding year is insufficient to cover the overhead, then no surplus remains to finance the current year's production. The system goes bankrupt. So long as each dual program in the sequence is feasible in the above sense, then optimal solution values $x_1(t)$, $x_2(t)$, $u_1(t)$ and $u_2(t)$ exist at each time t. When this occurs, $I(t) = I_D(t)$ by the duality theorem of LP.

It is important to emphasize that the solution values $x_1(t)$, $x_2(t)$, $u_1(t)$ and $u_2(t)$ describe and do not prescribe behavior in our model. That is, these values are not necessarily optimal ex post. For the industry as a whole, the actual optimum is the monopoly solution given by the quadratic programming problem[4] that arises when a perfect knowledge of the demand curves (Equation A3.27) is accounted for by the decision makers over time.

B. Phase Solutions

If a solution of the problem at the time t is unique, it lies at an extreme point of the LP feasible region. If there is more than one solution, then extreme point solutions are among them. Consequently, a solution at any time t can be represented by a set of equated constraints corresponding to the duality conditions. These equations give the algebraic description of the extreme point. Because of the recursive character of the sequence of programs, these sets of equated constraints identify difference equations that describe the behavior of the production variables and marginal value for a given time period. Since these sets may change from time to time, the system as a whole is a multiple-phase system. Questions concerning growth, cycles and equilibrium consequently boil down to an analysis of the conditions for the occurrence of specific phases (sets of difference equations) and the dynamic properties of each phase.

By examining the extreme solution possibilities we find the following six cases:

Phase 0 (Null phase): nothing produced, no imputed value.

Phase 1-s: fixed-factor constraints equated, the first commodity produced.

Phase 2-s: fixed-factor constraints equated, the second commodity produced.

Phase 1-r: financial constraints equated, the first commodity produced.

Phase 2-r: financial constraints equated, the second commodity produced.

Phase 12-rs: both constraints equated, both commodities produced.

To each of these extreme solutions corresponds a set of equations:

Phase 0: $x_1(t) = 0, x_2(t) = 0, u_1(t) = 0$ and $u_2(t) = 0$

Phase 1-s: $x_1(t) = 1, x_2(t) = 0, u_1(t) = a_1 + b_1 x_1(t-1) - c_1, u_2(t) = 0$

Phase 2-s: $x_2(t) = 1, x_1(t) = 0, u_1(t) = A3 + b_2 x_2(t-1) - c_2, u_2(t) = 0$

Phase 1-r: $x_1(t) = (1/c_1)[\Sigma_i\{a_i + b_i x_i(t-1)\}x_i(t-1) - C_0], x_2(t) = 0,$
$u_2(t) = (1/c_1)[a_1 + b_1 x_1(t-1) - c_1], u_1(t) = 0$

Phase 2-r: $x_2(t) = (1/c_2)[\Sigma_i\{a_i + b_i x_i(t-1)x_i(\mathrm{t}-1) - C_0], x_1(t) = 0,$
$u_2(t) = (1/c_2)[a_2 + b_2 x_2(t-1) - c_2], u_1(t) =0$

Phase 12-rs: $$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \left(\frac{1}{c_1-c_2}\right)\begin{bmatrix} -c_2 & 1 \\ c_1 & -1 \end{bmatrix}\begin{bmatrix} 1 \\ \Sigma_i\{a_i + b_i x_i(t-1)\}x_i(\mathrm{t}-1) - C_0 \end{bmatrix},$$

$$\begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} = \left(\frac{1}{c_1-c_2}\right) - \begin{bmatrix} -c_2 & c_1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} a_1 + b_1 x_1(t-1) - c_1 \\ a_2 + b_2 x_2(t-1) - c_2 \end{bmatrix}$$

Which phase holds in a given time period depends on the solution in the preceding time period and the relative positions of constraints and objective functions that the preceding solution brings about. Various possibilities are shown to occur, including phase and production periodicity, convergence to a stationary state and so on. It is important to note that solutions are multi-phase; they satisfy different sets of dynamic equations at various times during their evolution. Recursive programming is the problem of optimizing an infinite set of recursively generated linear functionals subject to an infinite set of recursively generated linear constraints. This could be expressed as the search for a set of functions that satisfy a system of nonlinear simultaneous difference inequalities (duality conditions). Among the possible solutions of a given system are sometimes one that exhibit an optimality property, in other words, they converge to some desirable state. Notice this is not always the case in practice, and it is in this respect that recursive programming is different from dynamic programming.

IX. CONCLUDING REMARKS

We have illustrated in this appendix several examples of a Markovian system, including Poisson or random demand pattern and queuing. These are both time-dependent probabilistic processes, or what is commonly known as stochastic processes. An equally common Markovian system is regular dynamic programming (DP), which can be both a deterministic and probabilistic process. All the above mentioned processes can be described by state transition equations, characterized by memoryless properties. This property suggests that the entire history of the process can be encapsulated in the last state of the system. A hybrid of a Markovian system and DP is found in Markovian decision processes, in which a stochastic process is optimized through time. We bring out the stationarity property of these processes, which allows steady-state equations to be written. A distinction is made, however, regarding optimizing the expected reward over the complete time horizon of a system and the expected reward per time period. While both policies can be stationary, they are intrinsically different since the former is specified for the entire life span while the latter is by definition time-period dependent. Although less known outside the economic literature, recursive programming (RP) is an allied concept to dynamic programming. It is a robust solution algorithm for sequential processes. Unlike DP, however, RP yields only local optimum where it exists, since it lacks the Markovian properties that allow decomposition of the optimization procedure into stages.

Admittedly we have only touched on a few fundamentals in this appendix, but it serves as an adequate prerequisite to understanding much of the discussions on stochastic facility location problems, one of the main topics in this text. It also allows understanding of the optimality conditions for certain heuristics in simultaneous location routing models, particularly the Route Improvement Synthesis and Evaluation (RISE) algorithm contained in the software CD/DVD and described also in the "Location-Routing" chapter of Chan (2005).

ENDNOTES

[1] An explanation of goodness of fit and chi-square is contained in Appendix 3 on "Statistical Tools" and the "Descriptive Tools" chapter respectively.

[2] Note that the return function can also be a function of both the decision variable and the state variable, $r_k(x_k, S_k)$. This is the case when delay is measured by a rectangle such as DFCE rather than a line segment such as AE/2 in Figure A3.10.

[3] DP is used to solve a multi-period capacitated location problem in the "Facility Location" chapter under the "Long-run Location Production Allocation Problems" section of Chan (2005).

[4] A quadratic program has an objective function that is a quadratic function of the decision variables, subject to a set of linear constraints. An example is the quadratic assignment problem in Chapter 3. Also, a monopolistic market model is given in the "Alternative Models of Spatial Competition" section of the "Spatial Equilibrium" chapter in Chan (2005), showing a nonlinear objective function.

REFERENCES

Chan, Y. (2005). *Location. transport and land-use: Modelling spatial-temporal information.* Berlin and New York: Springer.

Cooper, R. B. (1980). *Introduction to queuing theory.* New York: Elsevier Science Publishing.

Day, R. H. (1973). "Recursive programming models: A brief introduction" In *Studies in economic planning over space and time,* edited by G. G. Judge and T. Takayama. Amsterdam: North-Holland and New York: American Elsevier Publishing Co.

Howard, R. (1960). *Dynamic programming and Markov processes.* Cambridge, Massachusetts: Technology Press of Massachusetts Institute of Technology.

Morlok, E. F. (1978). *Introduction to transportation engineering and planning.* New York: McGraw-Hill.

Winston, W. L. (1994). *Operations research: Applications and algorithms,* 3rd ed. Belmont, California: ITP Duxbury Press.

Appendix 4

Review of Some Pertinent Optimization Schemes

This appendix provides an optimization background necessary for a better appreciation of the pertinent chapters. These chapters include location and routing and other prescriptive models of spatial analysis.

I. LINEAR PROGRAMMING

Linear programming is a procedure used to arrive at the best solution for a set of linear algebraic inequalities and a linear objective function. It should be obvious from Chapter 4 that the graphical method for solving linear programs (LP) is limited to models of two decision variables. An algebraic procedure is needed to solve LPs with numerous variables and inequalities. Also, such an algebraic technique is conducive to computer programming. One algebraic technique for solving LPs is the simplex algorithm. A large number of software packages are available to perform the computation. Because the method has been around for quite some time, most have been refined for computational efficiency. Here we outline the basic concepts mainly for a more coherent discussion of more general and efficient procedures. These include relaxation and decomposition techniques, which form the thrust of this appendix.

A. Simplex Algorithm

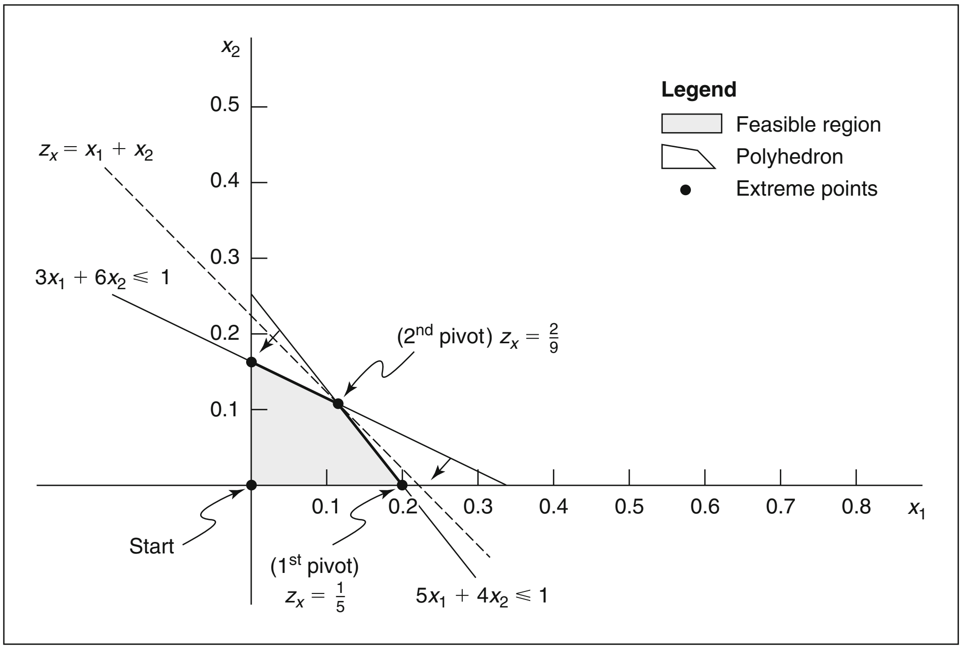
Consider the LP

$$\begin{aligned} &\text{Max } z_x = x_1 + x_2 \\ &s.t. \quad 3x_1 + 6x_2 \leq 1 \\ &\qquad\quad 5x_1 + 4x_2 \leq 1 \\ &x_i \geq 0 \quad i = 1, 2 \end{aligned} \tag{A4.1}$$

First, we solve this graphically as before in Figure A4.1, just for comparison with the algebraic method described below. To start the primal simplex algebraic procedure, the inequalities of the constraint equations are now changed into equalities by the addition of slack variables x_3 and x_4. The adjusted system of equations now looks like

$$\begin{aligned} z_x - x_1 - x_2 \qquad\qquad &= 0 \\ 3x_1 + 6x_2 \qquad + x_4 &= 1 \\ 5x_1 + 4x_2 + x_3 \qquad &= 1 \end{aligned} \tag{A4.2}$$

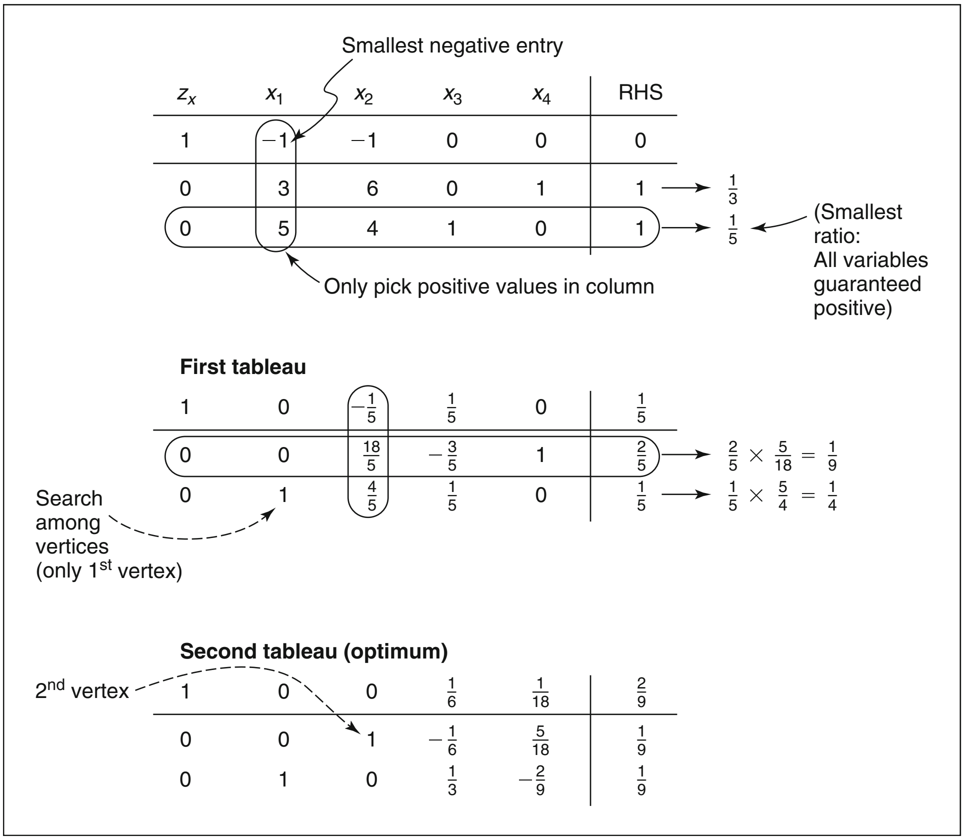
Figure A4.1 GRAPHIC SOLUTION TO LINEAR PROGRAM



where the right-hand side (RHS) of $(1, 1)^T$ represents positive resources to be allocated among the decision variables in order to maximize a figure of merit, called the objective function. This operation is an attempt to provide a starting basic feasible solution (BFS) to the LP. As it stands, a solution consisting of $(x_3, x_4) = (1, 1)$ and $(x_1, x_2) = (0, 0)$ is a perfectly good—albeit no where near optimal—solution to the LP. In this solution, $(x_3\ x_4)$ are the basic variables, and they form the basis of non-negative values. The variables $(x_1\ x_2)$ are nonbasic variables, assuming zero values. This BFS corresponds to the origin in the graphic plot shown in Figure A4.1.

This set of equations is then organized around a tableau, and Gaussian operations are performed as they would be in solving a set of simultaneous equations. As shown in Figure A4.2, the procedure pivots from one extreme point or vertex of the feasible region to another, corresponding to changing the basis in the algebraic context. Figure A4.2 is a good reference for the algebraic operations, showing the movement from extreme point to extreme point as the maximization objective function gets bigger and bigger. Two rules are followed in the pivoting operations. First, the column with the most negative number in the first row is picked to be the variable to enter the basis. For example, x_1 (or x_2) will be the variable to enter in the first pivot. This says that the best way to improve the value of the objective function is to go up the steepest slope, in this case along the x_1-axis. Because the terms have been moved from the right to the left in the objective function during the transformation shown above in Equation A4.2, only by engaging the variables with the negative cost coefficients will the figure of merit z increase. These are the variables j which will provide a net improvement in the objective function in the allocation of limited resources among several activity variables.

Figure A4.2 LINEAR PROGRAMMING TABLEAUX SHOWING PIVOTING OPERATIONS


Smallest negative entry
Only pick positive values in column
(Smallest ratio: All variables guaranteed positive)
Search among vertices (only 1st vertex)
2nd vertex

z_x	x_1	x_2	x_3	x_4	RHS	
1	-1	-1	0	0	0	
0	3	6	0	1	1	$\frac{1}{3}$
0	5	4	1	0	1	$\frac{1}{5}$

First tableau

1	0	$-\frac{1}{5}$	$\frac{1}{5}$	0	$\frac{1}{5}$	
0	0	$\frac{18}{5}$	$-\frac{3}{5}$	1	$\frac{2}{5}$	$\frac{2}{5} \times \frac{5}{18} = \frac{1}{9}$
0	1	$\frac{4}{5}$	$\frac{1}{5}$	0	$\frac{1}{5}$	$\frac{1}{5} \times \frac{5}{4} = \frac{1}{4}$

Second tableau (optimum)

1	0	0	$\frac{1}{6}$	$\frac{1}{18}$	$\frac{2}{9}$
0	0	1	$-\frac{1}{6}$	$\frac{5}{18}$	$\frac{1}{9}$
0	1	0	$\frac{1}{3}$	$-\frac{2}{9}$	$\frac{1}{9}$

Such net improvement is often referred to as the reduced cost, $(z_j - c_j)$. It signifies that the engagement of x_j will benefit the objective function by c_j but at an opportunity cost of z_j, reflecting idling other activities x_k $(k \neq j)$.

Once the entering variable has been identified, entries in the column directly under the entering variable (x_1, in this case) will be paired against the RHS. A ratio of the RHS entries and the positive elements in the column will be taken. The row with the smallest ratio will determine the variable to exit the basis. In the first pivot, it is clear that the third row, corresponding to x_3 will exit. This second rule keeps the next solution within the feasible region and also ensures a non-negative solution. The row with the smallest ratio identifies the most confining resource—among the two resources in this example—and keeps us operating "within our means". Choosing only the positive entries in the column of the entering variable is an attempt to stay away from unboundedness—in other words, the endless engagement of an activity variable since it consumes no real resources during its deployment.

Once the column and row have been identified, in this case column 2 and row 3, the element that belongs to both the row and column becomes the pivot. Gaussian operations are performed between all the rows to make this pivot unity in value and the rest of the entries in the column zero. Following our example, we have successfully carried this out in the first tableau. This tableau has a basis made up of x_1 and x_4 recognize that variables outside the basis (the nonbasic variables) are at zero value by definition, corresponds to the extreme point $(1/5, 0)^T$ in Figure A4.1. (Notice that should we pick x_2 as the variable to enter, the basis would be x_2 and x_3 and would correspond to the extreme point $(0, 1/6)^T$. Instead of being at "sea level" ($z_x = 0$) when we started at original vertex (0, 0), the new altitude at $(1/5, 0)^T$ is now 1/5.

A second pivot will introduce x_2 into the basis and elevate the altitude further to 2/9 at the new vertex (1/9, 1/9). We know we have arrived at the optimum since none of the entries in the first row are negative any more. Should we engage any of the variables to enter the basis, we will be descending, instead of ascending, the slope. In fact, the disappearance of all negative entries in the first row constitutes the termination rule. The readers probably recognize that these pivots are essentially effected by basis inverses in linear algebra. Matrix formulation of pivots will be shown below.

B. Some Other Key Concepts

The simplex procedure works because there is a finite number of extreme points, a convex combination of which will define all points in the feasible region, or polyhedron, of the LP. In other words, if $\mathbf{x}^1$ and $\mathbf{x}^2$ are two extreme points, the convex combination $w\mathbf{x}^1 + (1 - w)\mathbf{x}^2$ $(0 \leq w \leq 1)$ forms a point that will be within the polyhedron. Furthermore, an optimum has to occur at an extreme point. This is clear from the graphical plot of this LP example in Figure A4.1, in which the reader is challenged to show otherwise, excepting for the case when the objective function parallels one of the constraints. The same concept can be demonstrated in the unbounded polyhedron shown in the dual space of this LP.[1] In Figure A4.3, we have extreme directions $\mathbf{d}^1$ and $\mathbf{d}^2$ in addition to extreme points $\boldsymbol{\lambda}^1$, $\boldsymbol{\lambda}^2$ and $\boldsymbol{\lambda}^3$. It is clear from this figure that we can represent every point in the set as a convex combination of the extreme points plus a non-negative linear combination of the extreme directions. Consider the point $\boldsymbol{\lambda}$, which can be represented as $\boldsymbol{\lambda}^0$ plus a positive multiple of the extreme direction $\mathbf{d}^1$. Notice that $\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}$ points in the direction $\mathbf{d}^1$. But $\boldsymbol{\lambda}^0$ itself is a convex combination of the extreme points $\boldsymbol{\lambda}^1$ and $\boldsymbol{\lambda}^3$. Hence $\boldsymbol{\lambda}^0 = \boldsymbol{\lambda} + \mu\mathbf{d}^1 = w\boldsymbol{\lambda}^1 + (1 - w)\boldsymbol{\lambda}^3 + \mu\mathbf{d}^1$ where $0 \leq w \leq 1$ and $\mu \geq 0$.

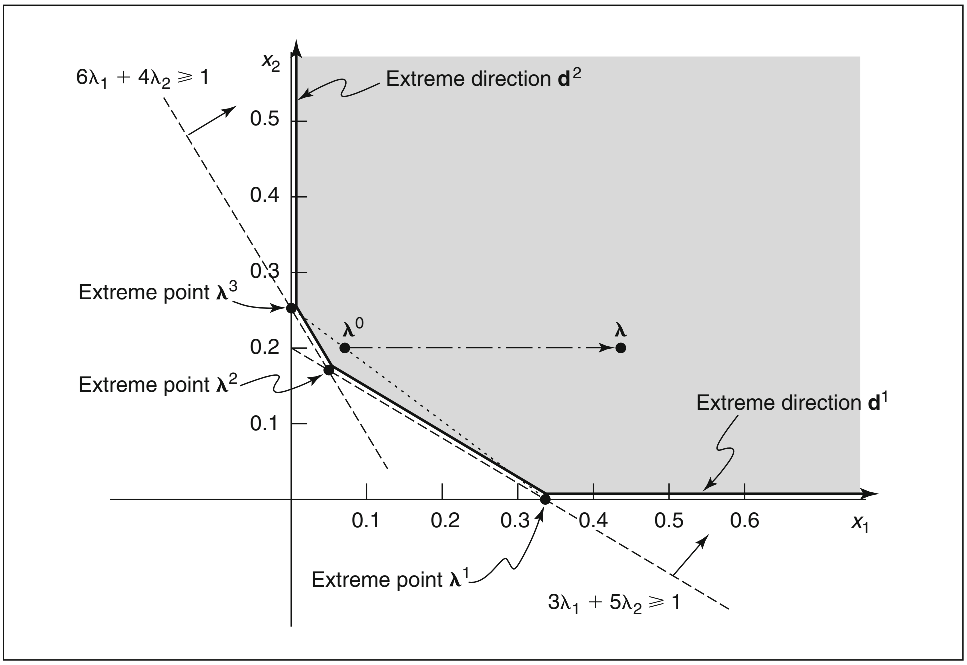
Notice this version of the simplex algorithm works only for these specific conditions:

- **(a)** maximization of the objective function,
- **(b)** less-than-or-equal-to in all the constraint equations, with positive RHSs,
- **(c)** non-negativity in all the decision variables.

But it is also a general procedure since many LPs can be cast into this form. For example, any minimization objective function can be converted to a maximization format:

$$\text{Min } z = \sum_{j=1}^{n} c_j\, x_j \text{ becomes Max } z' = \sum_{j=1}^{n} c_j\, x_j$$

Figure A4.3 DUAL OF LINEAR PROGRAMMING EXAMPLE



(Such a conversion suggests that the termination rule for a minimization LP should be when the first row of the simplex tableau consists of all non-positive entries.) Similarly, constraints not in the correct form can be converted:

$$\sum_j a_{ij}\, x_j \leq b_i \text{ becomes } \sum_j -a_{ij}\, x_j \leq -b_i$$

Equalities can be expressed in terms of two inequalities. Thus

$$\sum_j a_{ij}\, x_j = b_I \text{ becomes } \sum_j a_{ij}\, x_j \leq b_I \text{ and } \sum_j a_{ij}\, x_j \geq b_i$$

Finally, a negative RHS can be converted to a positive value by multiplying both sides of the constraint by a negative one. The simplex algorithm proceeds similarly as in the previous case once the model is cast into the form shown in Equation A4.1. Notice that there is really no magic in the form encapsulated in Equation A4.1. It is simply a convenient way to obtain a BFS, and it also allows us to explain the logic behind the simplex steps easily in terms of resource allocation. Should there be any difficulty in this conversion—and it will arise sometimes—other means for obtaining an initial BFS are necessary. There are several ways to do so, including solving the dual instead of the primal problem, wherein a minimization problem is converted to a maximization problem. An artificial variable may be added (with zero cost coefficient) to an equality constraint to make up the full rank of a BFS. Another way is to look for an initial solution based on existing operating conditions in practice. Interested readers may wish to consult Winston (1994) or Hillier and Lieberman (1990) regarding the precise procedures for doing this. In most applications, however, software packages are available to take care of the entire computational procedure, leaving more time for the user to formulate the LP and perform the analysis and to discern abnormalities in the computation when one occurs.

C. Theory of Simplex

We can summarize the simplex method as a set of matrix operations. First, we cast Equation A4.2 into an equation form:

$$\begin{aligned}&\text{Min } z \\ &\text{s.t.}\quad z - \mathbf{c}_B^T\mathbf{x}_B - \mathbf{c}_N^T\mathbf{x}_N = 0 \qquad \text{(A4.3)}\\ &\qquad\quad \mathbf{A}_B\mathbf{x}_B + \mathbf{A}_N\mathbf{x}_N = \mathbf{b} \\ &\qquad\quad \mathbf{x}_B,\ \mathbf{x}_N \geq \mathbf{0} \qquad \text{(A4.4)}\end{aligned}$$

Equation A4.4 is transformed in pivoting operations by pre-multiplying by $\mathbf{A}^{-1}_{B}$

$$\mathbf{x}_B + \mathbf{A}^{-1}_{B}\mathbf{A}_N\mathbf{x}_N = \mathbf{A}^{-1}_{B}\mathbf{b} \qquad \text{(A4.5)}$$

Multiplying Equation A4.5 by $\mathbf{c}_B$ and adding to Equation A4.3:

$$z + \mathbf{0x}_B + (\mathbf{c}_B^T\mathbf{A}^{-1}_{B}\mathbf{A}_N - \mathbf{c}_N)\mathbf{x}_N = \mathbf{c}_B\mathbf{A}^{-1}_{B}\mathbf{b} \qquad \text{(A4.6)}$$

By setting the nonbasic variables to zero $\mathbf{x}_N = 0$, Equation A4.5 yields $\mathbf{x}_B = \mathbf{A}_B^{-1}\mathbf{b}$ and Equation A4.6 yields $z = \mathbf{c}_B\mathbf{A}_B^{-1}\mathbf{b}$. The tableau looks like the following in each iteration, including the last and optimal iteration:

	z	$\mathbf{x}_B$	$\mathbf{x}_N$	RHS	
Row 0	1	$\mathbf{0}$	$\mathbf{c}_B^T\mathbf{A}^{-1}_{B}\mathbf{A}_N - \mathbf{c}_N^T$	$\mathbf{c}_B^T\mathbf{A}^{-1}_{B}\mathbf{b}$	(A4.7)
Row 1→m	$\mathbf{0}$	$\mathbf{I}$	$\mathbf{A}^{-1}_{B}\mathbf{A}_N$	$\mathbf{A}^{-1}_{B}\mathbf{b}$	

Consult the example worked out below for an illustration. To do this it is convenient to rearrange the slack variables in Equation A4.2 in terms of an identity matrix **I** by reversing rows 1 and 2 in the constraints.

Example

Given the following maximization LP tableau that has been rearranged into the format of Equation A4.3, where the basis $\mathbf{A}_B$ is the identity matrix:

z	x_3	x_4	x_2	x_1	RHS
1	0	0	−40	−10	0
0	1	0	1	1	10
0	0	1	5	2	30

Here $\mathbf{c}_B = (0\ 0)^T$, $\mathbf{c}_N = (40\ 10)^T$, $\mathbf{A}_N = \begin{bmatrix} 1\,1 \\ 5\,2 \end{bmatrix}$ and $\mathbf{b} = (10\ 30)^T$. The reduced cost for column 2 is $\mathbf{c}^T_B\mathbf{A}^{-1}_B\mathbf{A}_N(x_2) - \mathbf{c}(x_2) = 0 - 40 = -40$, which constitutes the pivot column to enter the basis. Now $\mathbf{A}_B = \begin{bmatrix} 1\,1 \\ 0\,5 \end{bmatrix}$, $\mathbf{c}_B = (0\ 40)^T$, $\mathbf{c}_N = (0\ 10)^T$, and $\mathbf{A}_N = \begin{bmatrix} 0\,1 \\ 1\,2 \end{bmatrix}$. According to the format of Equation A4.7, the inverse of the basis $\mathbf{A}_B$ is taken, and the refreshed tableau becomes

z	x_3	x_2	x_4	x_1	RHS
1	0	0	8	6	240
0	1	0	$-1/5$	$3/5$	4
0	0	1	$1/5$	$2/5$	6

This illustrates one pivot of the simplex. If Equations A4.3 and A4.4 are viewed as the initial tableau where $\mathbf{A}_N$ is an identity matrix and Equation A4.7 the optimal tableau (as in a 2 × 4 example tableau), then $\mathbf{c}_N$ is a zero vector and $\mathbf{c}^T_B\mathbf{A}_B^{-1}\mathbf{A}_N = \mathbf{c}^T_B\mathbf{A}^{-1}_B$ is simply the dual vector. In general, $\lambda_i = \mathbf{c}_B\mathbf{A}^{-1}_B\mathbf{A}_N(s_i\ ) - \mathbf{c}_N(s_i)$ for the nonbasic variable s_i and zero for the basic variables, where s_i is the slack for inequality i. Thus in this example, the dual vector can be read from the first row of the last tableau, directly above the identity matrix where the slack variables were in the initial tableau. This means $\lambda_1 = 0$ and $\lambda_2 = 8$ as the readers can verify. Furthermore,$\mathbf{A}_B^{-1}$ can be found where the identity matrix forthe slacks was, namely $\mathbf{A}_B^{-1} = \begin{bmatrix} 1 & -1/5 \\ 0 & 1/5 \end{bmatrix}$. ■

II. NETWORK-WITH-SIDE-CONSTRAINTS

While the simplex procedure is a good way to introduce optimization procedures, there are more efficient techniques to solve such a model, depending on the structure of the tableau. Network-flow programming is an excellent way to attack large-scale models, when the tableau can be cast into special formats.[2] A generalized network-flow algorithm is network-with-side-constraints (NSC), which can be applied toward problems having the following structure:

$$\begin{array}{rll} \text{Min} & \mathbf{w}^T\mathbf{x} + \mathbf{c}^T\mathbf{y} & \\ \text{s.t.} & \mathbf{A}\mathbf{x} & = \mathbf{b} \\ & \mathbf{B}\mathbf{x} \ + \mathbf{C}\mathbf{y} & = \mathbf{b}' \end{array} \qquad \text{(A4.8)}$$

where $\mathbf{A}$ is the network matrix, $\mathbf{B}$ and $\mathbf{C}$ are arbitrary matrices. $\mathbf{d}$ and $\mathbf{d}'$ are arc capacity vectors on flow variables $\mathbf{x}$ and other variables $\mathbf{y}$ respectively. The algorithm takes advantage of the nice properties of the network matrix $\mathbf{A}$, which is assumed to be the more prominent part of the tableau in comparison to $\mathbf{B}$ and $\mathbf{C}$, and achieves computational efficiency that way. Let us call NETSIDE the off-the-shelf program available in SAS/OR, CPLEX, and other production codes to solve NSC problems.

A. Multicommodity-Flow Problem

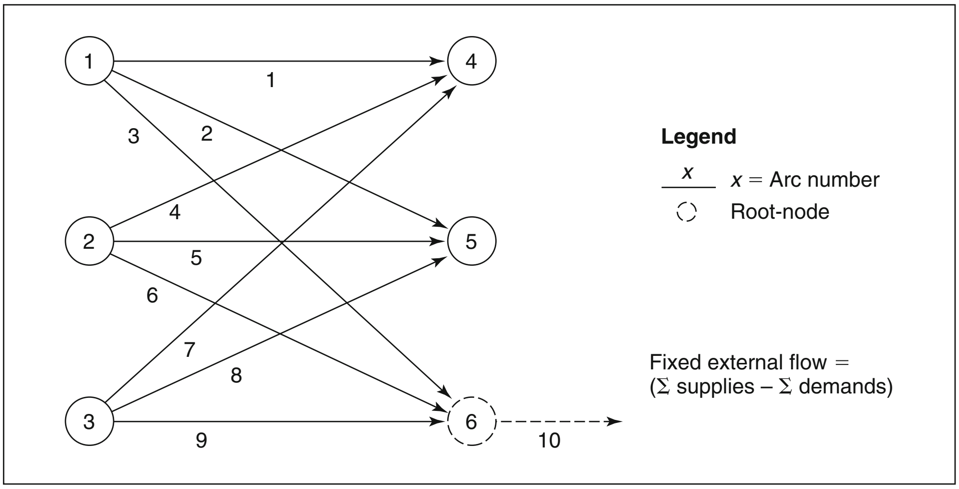
A good way to illustrate NSC is through a well-known special case: the multicommodity-flow problem. Consider the two-commodity ($r = 1, 2$) problem illustrated in Figure A4.4, where the supply for nodes 1, 2 and 3 for both commodities is at most 2, and the demand at nodes 4, 5, and 6 for both commodities is at least 2. The individual bounds d_k^r are infinite for all arcs and both commodities ($r = 1, 2$), and the mutual capacity for arc 1 (d_1) is 2 and all other arcs (d_j, $j \neq 1$) have a capacity of 3. A corresponding tableau for this problem is shown in Figure A4.5. It is clear that the tableau can be partitioned into **A**, **B**, and **C** matrices, with **A** being the block-diagonal network matrix containing the two commodities, while **B** and **C** constitute the arc flow constraints, **x** is the regular network flow vector and **y** the slack flow vector.

Specialization of the primal simplex algorithm for network programs results in the simplex-on-a-graph algorithm, where there is a graphic/labeling replacement for each step of the simplex. The tableau with a basic solution looks like Figure A4.5, where the basic variables are boxed within the solid lines and the nonbasic variables to enter the basis are housed in dashed lines. The process in which a nonbasic variable enters the basis, or the incremental method of inverting a basis, is performed by an orientation sequence. For example, see the first column of the tableau in Figure A4.5, where the NETSIDE algorithm (Kennington and Helgason 1980) is being illustrated. Notice arc 4 of commodity 1 is introduced into the basis by an orientation sequence. The example illustrates this algorithm in the tableau, where primal feasibility is maintained and complementary slackness relaxed.[3]

By way of a definition, the orientation sequence of a path P of length k, $O'(P)$, is specified by a sequence of these numbers:

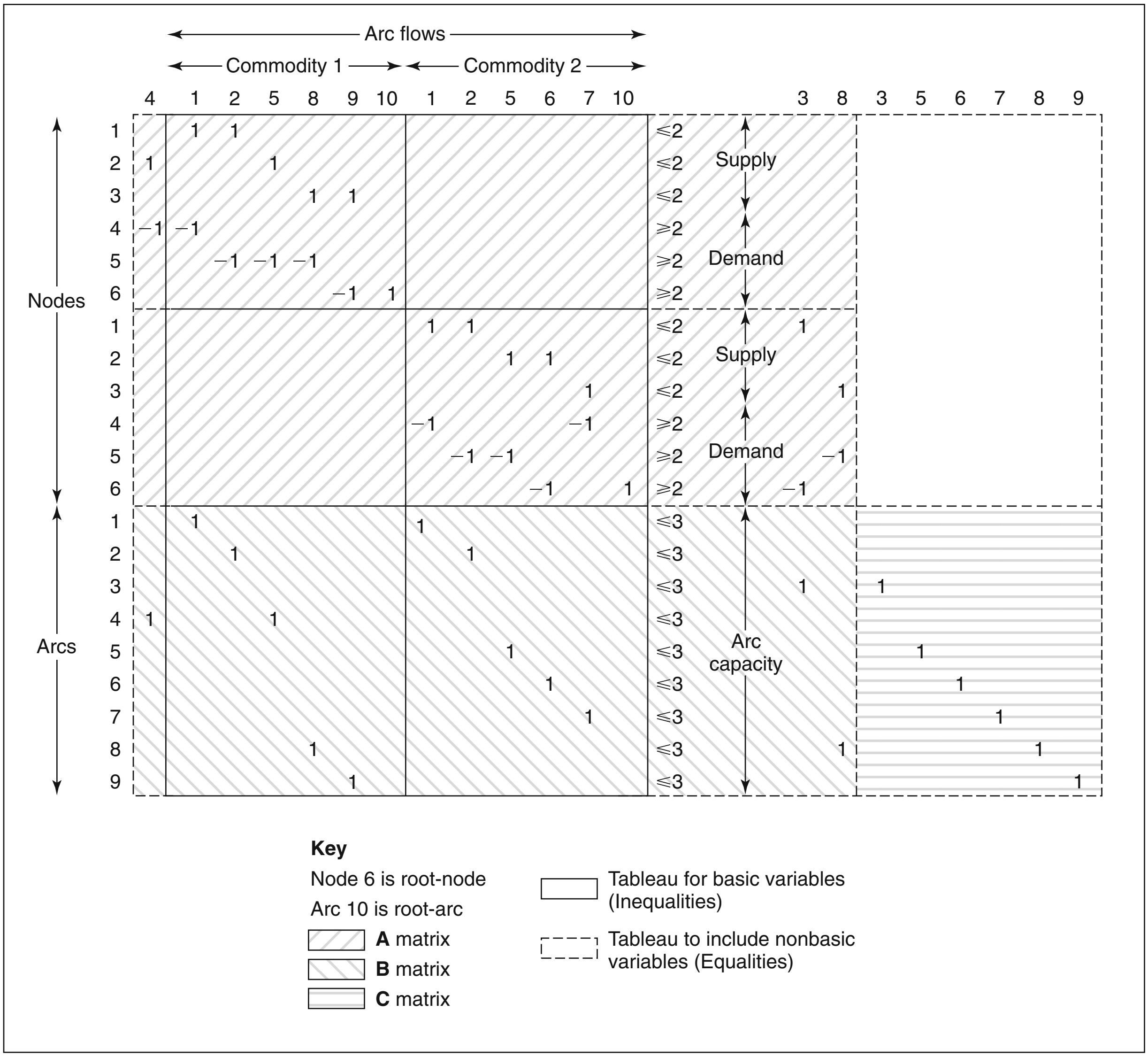
$$O'_i(P) = \begin{cases} +1 & \text{if } e_{j_i} = (i, i+1) \\ -1 & \text{if } e_{j_i} = (i+1, i) \end{cases} \quad i = 1, \ldots, k \tag{A4.9}$$

Figure A4.4 SAMPLE MULTICOMMODITY-FLOW NETWORK



SOURCE: Kennington and Helgason (1980). Reprinted with permission.

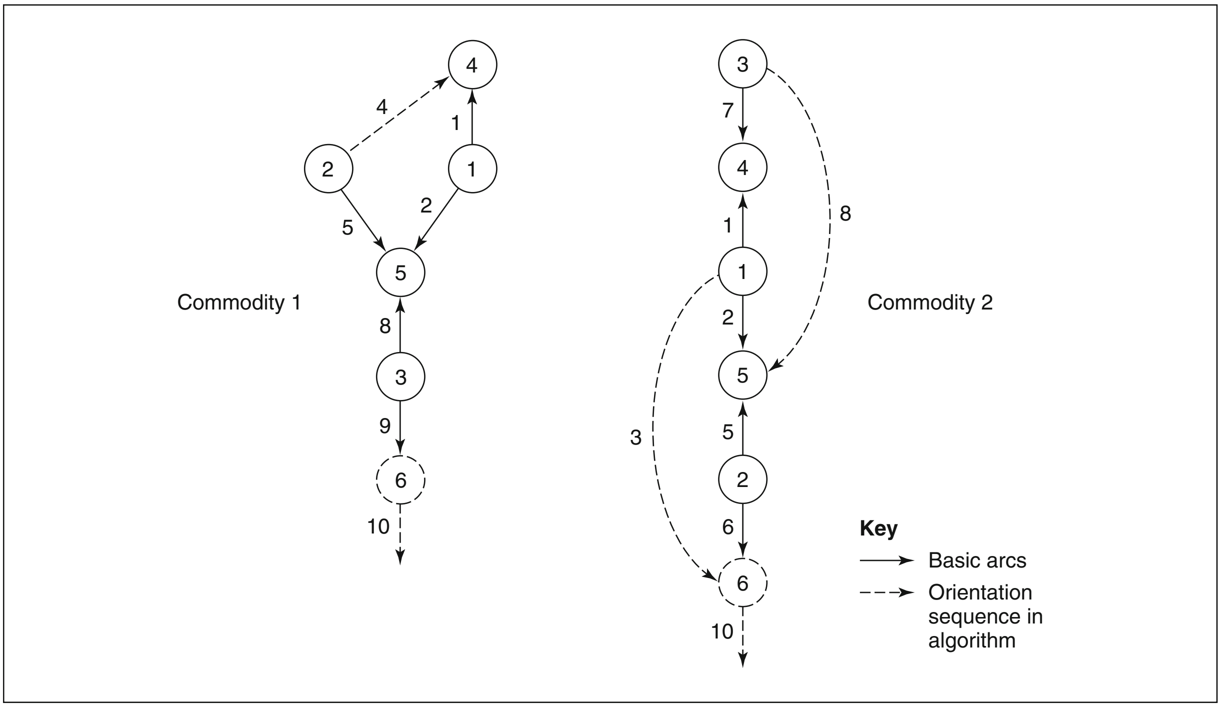
Figure A4.5 BLOCK DIAGONAL MATRIX CORRESPONDING TO AN ORIENTATION SEQUENCE



		4	1	2	5	8	9	10	1	2	5	6	7	10		3	8	3	5	6	7	8	9
Nodes	1		1	1											⩽2								
	2	1			1										⩽2								
	3					1	1								⩽2								
	4	−1	−1												⩾2								
	5			−1	−1	−1									⩾2								
	6						−1	1							⩾2								
	1								1	1					⩽2	1							
	2										1	1			⩽2								
	3												1		⩽2		1						
	4								−1				−1		⩾2								
	5									−1	−1				⩾2		−1						
	6											−1		1	⩾2	−1							
Arcs	1		1						1						⩽3								
	2			1						1					⩽3								
	3														⩽3	1		1					
	4	1			1										⩽3								
	5										1				⩽3				1				
	6											1			⩽3					1			
	7												1		⩽3						1		
	8					1									⩽3		1					1	
	9						1								⩽3								1

where e_{j_i} is the arc j associated with node/vertex i. The associated basis trees for the tableau in Figure A4.5 are sketched in Figure A4.6 where the nonbasic network flow variables are arcs 4, 3, and 8. An example of the orientation sequence for $P = \{3, 5, 2, 6\}$ is $O'(P) = \{1, -1, 1\}$. In household terms, the orientation sequence records whether the path is with the direction of the arrow or against the direction of the arrow. This is recorded using a + 1 and −1 respectively. The orientation sequence $O'(P)$ performs a series of computations on the graph similar to basis inversion in simplex. In Figure A4.6, we can show how a nonbasic vector in the commodity-1 tree can be represented in terms of basic vectors by way of an orientation sequence. For example, the nonbasic arc-4

Figure A4.6 BASIS TREES SHOWING ORIENTATION SEQUENCE



SOURCE: Kennington and Helgason (1980). Reprinted with permission.

considered for entering the network basis can be represented in terms of three basic arcs 5, 2 and 1, and the algebra proceeds as follows:

$$
\begin{aligned}
\mathbf{A}(4) &= \mathbf{A}(5) - \mathbf{A}(2) + \mathbf{A}(1) \\
&= \mathbf{e}^{2(4)} - \mathbf{e}^{5(4)}) - (\mathbf{e}^{1(4)} - \mathbf{e}^{5(4)}) + (\mathbf{e}^{1(4)} - \mathbf{e}^{4(4)}) \\
&= \mathbf{e}^{2(4)} - \mathbf{e}^{4(4)} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \\ 0 \\ 0 \end{bmatrix}
\end{aligned} \tag{A4.10}
$$

Here $\mathbf{A}(j)$ stands for the column vector in the simplex tableau for arc j, $\mathbf{e}^{i(j)}$ is the unitary column vector for arc j with the unitary entry in the ith row. Care should be exercised in distinguishing the arc notation e_{j_i} from the unitary column notation $\mathbf{e}^{i(j)}$.

B. The Network-with-Side-Constraints Algorithm

Having illustrated some basic ideas, we will show an NSC algorithm—NETSIDE—step by step through an example (Kennington and Helgason 1980). The following problem is of the same format as Equation A4.8:

$$
\begin{array}{lccccccccccccr}
\text{Min} & & & 10x_2 & +2x_3 & & +3x_5 & +4x_6 & & & & & \\
\hline
\text{s.t.} & x_7 & +x_1 & +x_2 & & & & & | & & & = & 10 \\
 & & -x_1 & & +x_3 & +x_4 & -x_5 & & | & & & = & 0 \\
 & & & -x_2 & -x_3 & & +x_5 & +x_6 & | & & & = & 0 \\
 & & & & & -x_4 & & -x_6 & | & & & = & -10 \\
\hline
 & & 10x_1 & & -2x_3 & +3x_4 & -2x_5 & & | & +y_1 & & = & 16 \\
 & x_1 & & & +4x_3 & & +x_5 & & | & & +y_2 & = & 10 \\
\hline
\end{array}
\qquad \text{(A4.11)}
$$

Recall that regular LP simplex starts with a full rank m for the constraint matrix, where m is the number of nodes in the node-arc incidence matrix. But in Chapter 4, we suggested that the rank of the node-arc incidence matrix **A** is of rank $m - 1$. An artificial variable is added to make up the full rank of m as required in LP. An artificial variable is added to one of the nodes, say node m. The augmented constraint matrix now looks like $[\mathbf{A}, \mathbf{e}^{\mathbf{m(m)}}]$, with the additional unitary column vector corresponding to the mth root-arc in a basis of $(m - 1)$ arcs. This arc is added to root-node m. Since any basic LP solution must contain m linearly independent columns, the artificial variable must appear in every basic simplex-on-a-graphsolution (in other words, every tree). An artificial variable in LP is added to an equality constraints in LP merely to provide a starting BFS; correspondingly this artificial arc carries with it a zero cost. We have already seen examples of a root arc, namely arc 10, and root node 6 in Figure A4.4 through Figure A4.6. More examples will be forthcoming in the following computation steps. For the current example (Figure A4.7), all variables are bounded as indicated in the table below, with x_7, the root arc flow, bounded between 0 and ∞, and the slack flows also uncapacitated $0 \leq y_j \leq \infty$.

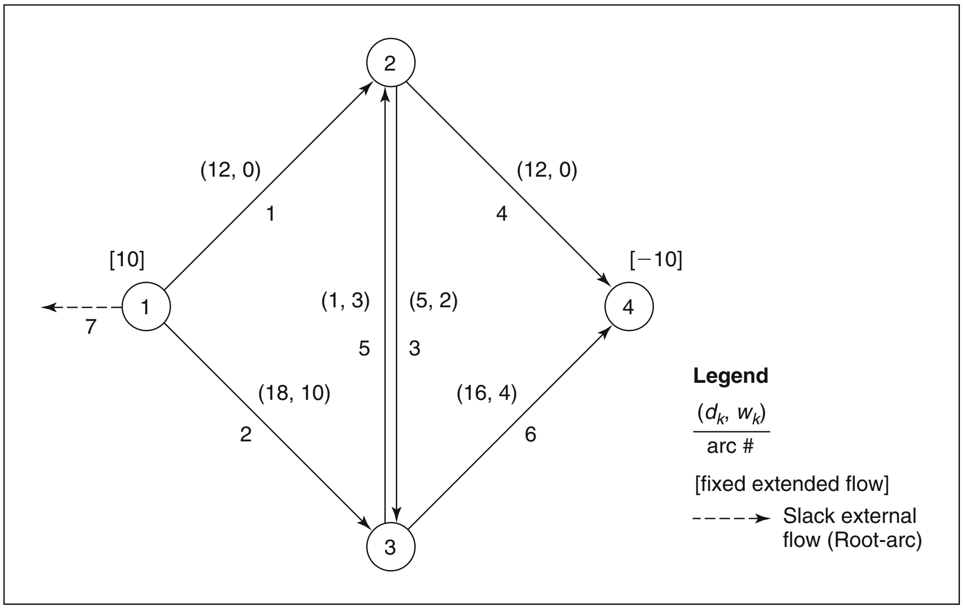
Arc j	1	2	3	4	5	6
d_j	12	18	5	12	1	16

1. Initialization Step. Equation A4.11 shows that the tableau for the example problem can be partitioned into matrices **A**, **B**, and **C** as was the case with the multicommodity flow problem. The initial basis for this tableau is a 6 × 6 matrix $\overline{\mathbf{B}}$.

The algebraic representation is the matrix $\overline{\mathbf{B}} = \begin{bmatrix} \mathbf{G} & \mathbf{H} \\ \mathbf{D} & \mathbf{F} \end{bmatrix}$ which is

x_7	x_2	x_5	x_6	x_1	x_3
1	1			1	
		–1		–1	1
	–1	1	1		–1
			1		
		–2		10	–2
		1		1	4

Figure A4.7 EXAMPLE TO ILLUSTRATE NETWORK-WITH-SIDE-CONSTRAINT ALGORITHM



where **G** is a square sub-matrix of the network matrix **A** carrying the same rank, with det(**G**) ≠ 0, constituting a rooted spanning tree. The corresponding graphical representation is a tree, which is shown in Figure A4.8.

Inverse of this matrix takes on special form, requiring only two inverses $\mathbf{G}^{-1}$ and $\mathbf{Q}^{-1}$:

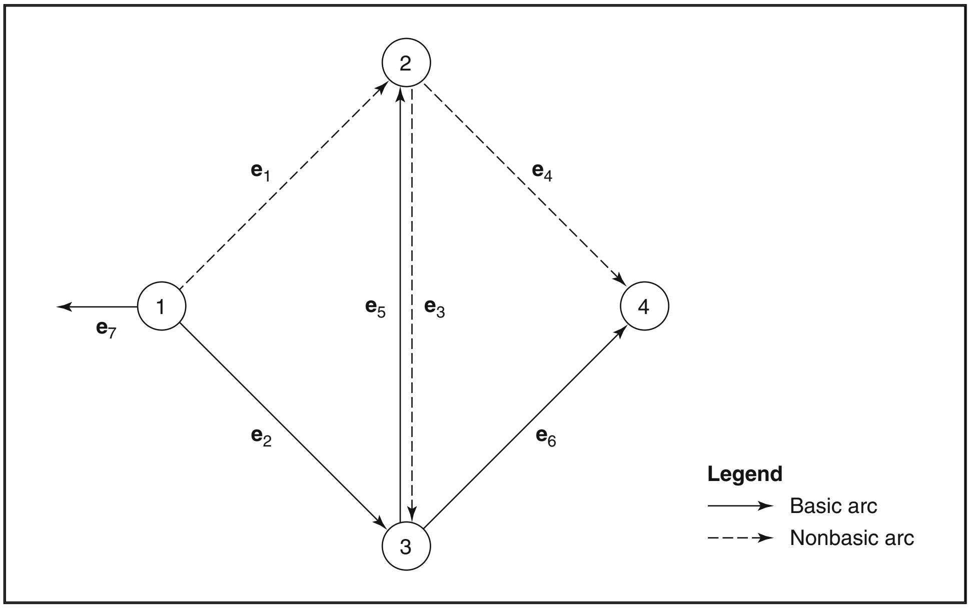
$$\overline{\mathbf{B}}^{-1} = \begin{bmatrix} \mathbf{G}^{-1} + \mathbf{G}^{-1}\mathbf{H}\mathbf{Q}^{-1}\mathbf{D}\mathbf{G}^{-1} & -\mathbf{G}^{-1}\mathbf{H}\mathbf{Q}^{-1} \\ -\mathbf{Q}^{-1}\mathbf{D}\mathbf{G}^{-1} & \mathbf{Q}^{-1} \end{bmatrix} \quad i = 1, \ldots, k \qquad \text{(A4.12)}$$

where $\mathbf{Q}^{-1} = \mathbf{F} - \mathbf{D}\mathbf{G}^{-1}\mathbf{H}$. Even though the $\mathbf{G}^{-1}$ can be arrived at by graphical means, we show the straightforward inversion below as a start:

$$\mathbf{G}^{-1} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -1 & -1 & -1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \quad \mathbf{Q}^{-1} = \begin{bmatrix} & 1/12 & 1/15 \\ 0 & 1/5 & \end{bmatrix} \text{and the initial } \mathbf{B}^{-1} \text{ looks like}$$

1	1	1	1		
	−9/10	−1	−1	−1/12	−1/15
	−7/10			−1/12	−2/15
			−1		
	−1/10			1/12	1/15
	1/5				1/5

Figure A4.8 TREE REPRESENTING INITIAL BASIS



The current solution, as evaluated by $\mathbf{B}^{-1}\begin{bmatrix}\mathbf{b}\\ \mathbf{b}'\end{bmatrix}$, is

$$(x_1^B, x_2^B, x_3^B, x_4^N, x_5^B, x_6^B, x_7^B, y_1^N, y_2^N) = (2\ 8\ 2\ 0\ 0\ 1\ 0\ 0\ 0\ 0)$$

where the superscript B marks a basic variable and N a nonbasic variable.

After guessing at an initial basis $\overline{\mathbf{B}}$, and hence $\mathbf{G}$ (or the basis tree T_B) for $\mathbf{A}$, we proceed with the two basic steps of LP simplex: entry and exit of variables into $\mathbf{G}$. The pricing step selects the entry variable while the ratio test selects the exit variable.

2. Pricing Step. An integral part of obtaining the reduced costs in the top row of the simplex tableau is computing the dual variable. The dot product of the dual vector, representing nodal potentials or "odometer readings" v_j for network flow, and the column vector under consideration will obtain the reduced cost $\mathbf{v}^T\mathbf{A}_N(k) - w_k$ of the column concerned. Notice the column for the kth arc (i, j) is in the nonbasic matrix outside $\overline{\mathbf{B}}$. In a network, the reduced cost is the potential difference across the arc $(v_j - v_i)$ that overcomes the arc cost w_{ij}, $\overline{v}_{ij} = (v_j - v_i) - w_{ij}$. To obtain the dual variable v_j in a network with side constraints:

$$(\mathbf{v}^1 \,|\, \mathbf{v}^2)^T = (\mathbf{w}^1 \,|\, \mathbf{w}^2)^T\overline{\mathbf{B}}^{-1} = [\{(\mathbf{w}^1)^T + (\mathbf{w}^1)^T\mathbf{G}^{-1}\mathbf{H}\mathbf{Q}^{-1}\mathbf{D} - (\mathbf{w}^2)^T\mathbf{Q}^{-1}\mathbf{D})\mathbf{G}^{-1} \,|\, ((\mathbf{w}^2)^T - (\mathbf{w}^1)^T\mathbf{G}^{-1}\mathbf{H}\}\mathbf{Q}^{-1}]$$

according to Equation A4.12 distinguishing between the spanning tree and nonspanning tree parts of $\overline{\mathbf{B}}^{-1}$. We will compute this in several steps utilizing graphical means where feasible. In this effort, we utilize two well-known facts. The rows and columns of the node-arc incidence matrix of any spanning tree can be rearranged to be lower triangular. The converse is the well-known result that every basis matrix defines a spanning tree.

Step 1. First, we calculate $(\pi^1)^T = (\mathbf{w}^1)^T\mathbf{G}^{-1}$ as follows. Let $\mathbf{G}$ be the basis with corresponding basis tree T_B. By virtue of Equation A4.5, any the kth component of dual variable can be obtained by first solving $\mathbf{Gy}' = \mathbf{A}(k) = \mathbf{e}^i(k) - \mathbf{e}^j(k)$ for the updated column $\mathbf{y}'$. Here $\mathbf{e}^i(k)$ and $\mathbf{e}^j(k)$ are the unitary vectors for arc k made up of $+1$ and -1 in the columns of the node-arc incidence matrix corresponding to the beginning node of starting arc and the ending node of the terminating arc in a path P, as illustrated in Equation A4.10. The dual variable is simply $(\mathbf{w}^1)^T\mathbf{y}'$. The basis of a min-cost-flow network program $\mathbf{G}$ (or the tree T_B) can be put in lower triangular form with $+1$ or -1 on the diagonals. This means the system of equations $\mathbf{Gy}' = \mathbf{A}(k)$ can be solved by simple forward substitution process. Since $\mathbf{G}$ is triangular, $\mathbf{y}'$ may be obtained directly and hence algebraic inverse $\mathbf{G}^{-1}$ is not required. We further make use of T_B to solve this triangular system. Let $P = \{1, 2, \ldots, n + 1\}$ be the unique path in T_B linking node $i(k)$ to node $j(k)$, then

$$\sum_{i=1}^{n} O_i'(P)\mathbf{A}(j_i) = \mathbf{e}^{i(k)} - \mathbf{e}^{j(k)}$$

In other words, if the arcs in T_B are ordered as $\mathbf{e}_{k_1,}\mathbf{e}_{k_2}, \ldots, \mathbf{e}_{k_j}$ corresponding to the columns of $\mathbf{G}$, then the pth component of $\mathbf{y}'$ can be determined by the orientation sequence

$$y_p = \begin{cases} O_i'(P) & \text{if } e_{k_p} = e_{j_i} \in P \\ 0 & \text{otherwise} \end{cases} \tag{A4.13}$$

A clarification note is in order at this point. Decision variable $\mathbf{y}$ is to be distinguished from updated column $\mathbf{y}'$, and $\mathbf{A}(j)$ here refers to the jth column in $\mathbf{A}$.

Now the arcs in the tree T_B is already ordered in the columns of

$$\mathbf{G} = \begin{array}{c} \begin{array}{cccc} e_7 & e_2 & e_5 & e_6 \end{array} \\ \left[\begin{array}{rrrr} 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 1 & 1 \\ 0 & 0 & 0 & -1 \end{array}\right] \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}$$

which is triangular. For column 2 of the network matrix in Equation A4.11, we have a column vector $\mathbf{A}(2) = (1\ -1\ 0\ 0)^T = (1\ 0\ 0\ 0)^T - (0\ 1\ 0\ 0)^T$. This is converted to $\mathbf{y}'$ by mapping path $P = \{1, 3, 2\}$ in the tree T_B in Figure A4.8 against the orientation sequence, resulting in $(0\ 1\ 1\ 0)^T$ according to Equation A4.13. In other words, we go down the vector entries corresponding to nodes 1, 2, 3, and 4 and ask where the arrow on the arrival node points. Is it with the path orientation (hence a $+1$ is assigned) or is it against (hence a -1 is assigned)? Notice $\mathbf{y}'$ checks out the matrix inverse algebra of

$$\mathbf{G^{-1}A}(2) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -1 & -1 & -1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

as computed by "brute force" method above.

Now that the updated column $\mathbf{y}(2)$ in the network part of network-with- side-constraint tableau has been computed by graphical means as $(0\ 1\ 1\ 0)^T$, the dual variable associated with node 2—or the "odometer reading" at node 2—of the network $\pi^1_2 = (\mathbf{w}^1)^T\mathbf{y}(2)$ can readily be computed. This is done by summing up the mileage $-3 - 10 = -13$, using the convention that flow goes from the root node (0 potential) to lower potentials (negative "odometer readings") at the other nodes. More formally, the dual variable $(\mathbf{w}^1)^T\mathbf{y}'$ for each updated column $\mathbf{y}'$, or $(\mathbf{w}^1)^T\mathbf{G}^{-1}$, is simply

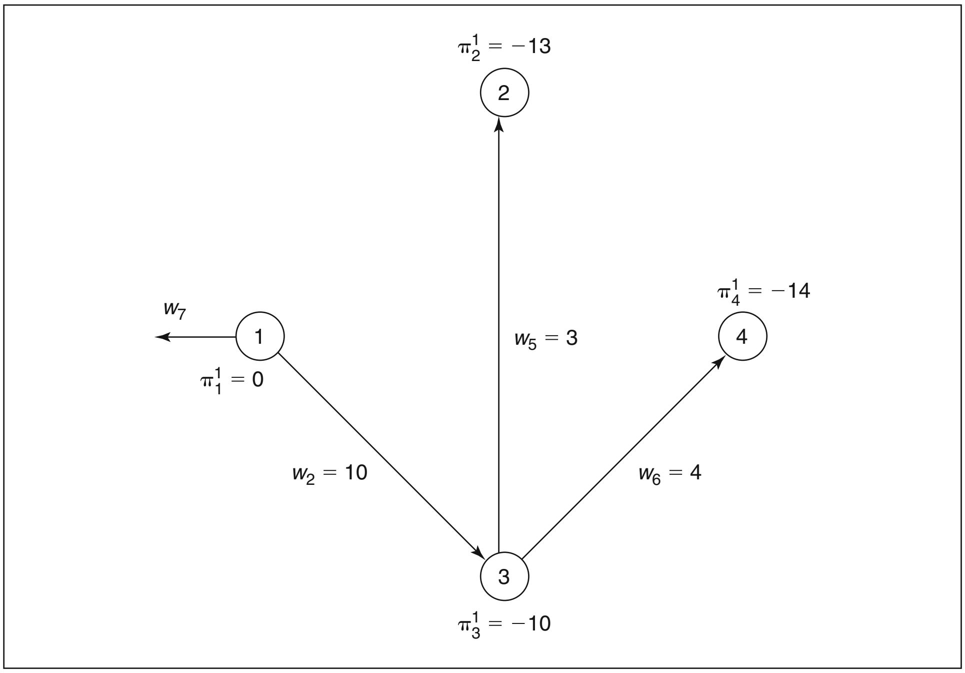
$$\pi_j = \sum_{i=1}^{n} w_{ji} O'_i(P)$$

where the orientation sequence $O'_2(P)$ *and* $O'_3(P)$ are taken as −1's, a convention which will be explained shortly. Thus the complete pricing vector $\boldsymbol{\pi}^1 = (\pi^1_1, \pi^1_2, \pi^1_3, \pi^1_4) = (0 - 13 - 10 - 14)^T$ can be represented in Figure A4.9, where the superscript 1 is the extra notation that identifies this as the duals associated with the network part of the network-with-side-constraint tableau.

Notice that in this figure, it can be deduced also from the sequence $x_7\ x_2\ x_5\ x_6$ in $\mathbf{G}$ that

$$\mathbf{G}^{-1} = \begin{array}{ccccc} 1 & 2 & 3 & 4 & \\ 1 & 1 & 1 & 1 & e_7 \\ 0 & -1 & -1 & -1 & e_2 \\ 0 & -1 & 0 & 0 & e_5 \\ 0 & 0 & 0 & -1 & e_6 \end{array}$$

Figure A4.9 CALCULATION OF $\boldsymbol{\pi}^1$ FOR PRICING



which checks out with the algebraic inverse shown earlier. Here the first row are all ones corresponding to the orientation sequence of the root node, which is taken as +1 by convention. The other arcs, if pointing in the opposite direction, would have a negative orientation sequence. This explains a dual variable π_2^1 of −13 instead of +13 and $\pi_3^1 = -10$ instead of +10.

Step 2. Going back to the matrix expression for the dual variables $(v^1 \mid v^2)$ at the beginning of this pricing step discussion. Let $\pi^2 = [(\mathbf{w}^1)^T + (\mathbf{w}^1)^T\mathbf{G}^{-1}\mathbf{H}\mathbf{Q}^{-1}\mathbf{D} - (\mathbf{w}^2)^T\mathbf{Q}^{-1}\mathbf{D}] = [(\mathbf{w}^1)^T + \pi^1\mathbf{H}\mathbf{Q}^{-1}\mathbf{D} - \mathbf{w}^2)^T\ \mathbf{Q}^{-1}\mathbf{D}]$. From this formula, we can compute $\pi^2 = (0\ 10\ 7/10\ 4)^T$ as:

$$(\boldsymbol{\pi}^2)^T = (0\ 10\ 3\ 4) + (0\ -13\ -10\ -14)\begin{bmatrix} 1 & 0 \\ -1 & 1 \\ 0 & -1 \\ 0 & 0 \end{bmatrix}\left[\begin{bmatrix} 1/12 & 1/15 \\ 0 & 1/5 \end{bmatrix}\right]\begin{bmatrix} 0 & 0 & -2 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{A4.14}$$

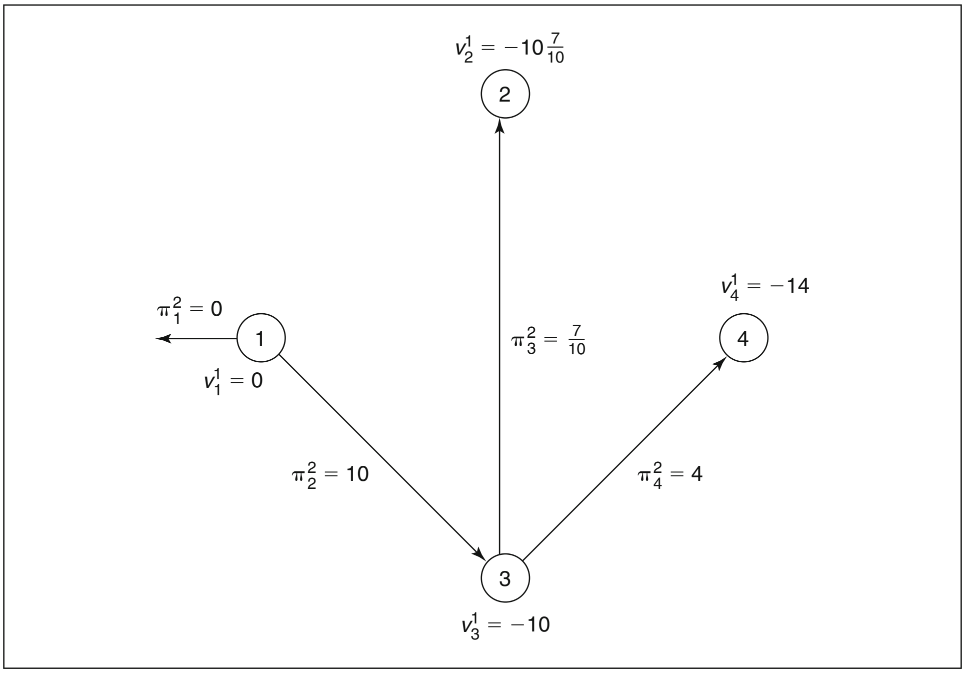
$$-(0\quad 2)\left[\begin{bmatrix} 1/12 & 1/15 \\ 0 & 1/5 \end{bmatrix}\right]\begin{bmatrix} 0 & 0 & -2 & 0 \\ 0 & 0 & 10 & 0 \end{bmatrix}$$

Step 3. $(\mathbf{v}^1)^T = (\pi^2)^T\mathbf{G}^{-1}$ is equivalent to step 1 in which π^2 replaces $\mathbf{w}^B$. This is again solved graphically in Figure A4.10 by computing nodal potentials, resulting in $v^1 = (v_1^1\ v_2^1\ v_3^1\ v_4^1)^T = (0\ -10\ 7/10\ -10\ -14)^T$.

Step 4. $(\mathbf{v}^2)^T = [(\mathbf{w}^2)^T - (\pi^1)^T\mathbf{H}]\mathbf{Q}^{-1}$

$$(\mathbf{v}^2)^T = \left\{(0\ 2) - (0\ -13\ -10\ -14)\begin{bmatrix} 1 & 0 \\ -1 & 1 \\ 0 & -1 \\ 0 & 0 \end{bmatrix}\right\}\left[\begin{bmatrix} 1/12 & 1/15 \\ 0 & 1/5 \end{bmatrix}\right] = \left(-1\ \tfrac{1}{12}\ \tfrac{1}{15}\right) \tag{A4.15}$$

Figure A4.10 CALCULATION OF V^1



We now calculate the reduced costs using the dual variables $(\mathbf{v}^1 \mid \mathbf{v}^2)^T$ by examining the nonbasic column under x_4, the nonbasic variable under consideration: $\mathbf{A}_N(4) = (0\ +1\ 0\ -1\ +3\ 0)^T$, where the reduced cost is $(\mathbf{v}^1 \mid \mathbf{v}^2)^T\mathbf{N}(4) - w_4$. For x_4: $v_2^1 - v_4^1 + 3v_1^2 - w_4 = -10\ 7/10 - (-14) + 3(-1\ 1/12) - 0 > 0$. This means entering of x_4 into the basis.

3. Ratio Test. Now is the time to pick an exit variable. Before the ratio test for exist variable selection is done, however, column updates need to be performed according to the revised simplex method, wherein only the columns of interest $\mathbf{y}'$ and $\mathbf{b}$ (i.e., column k of the tableau and the RHS) are updated. Here $\mathbf{y}' = \overline{\mathbf{B}}^{-1}\overline{\mathbf{A}}(k)$, where we write

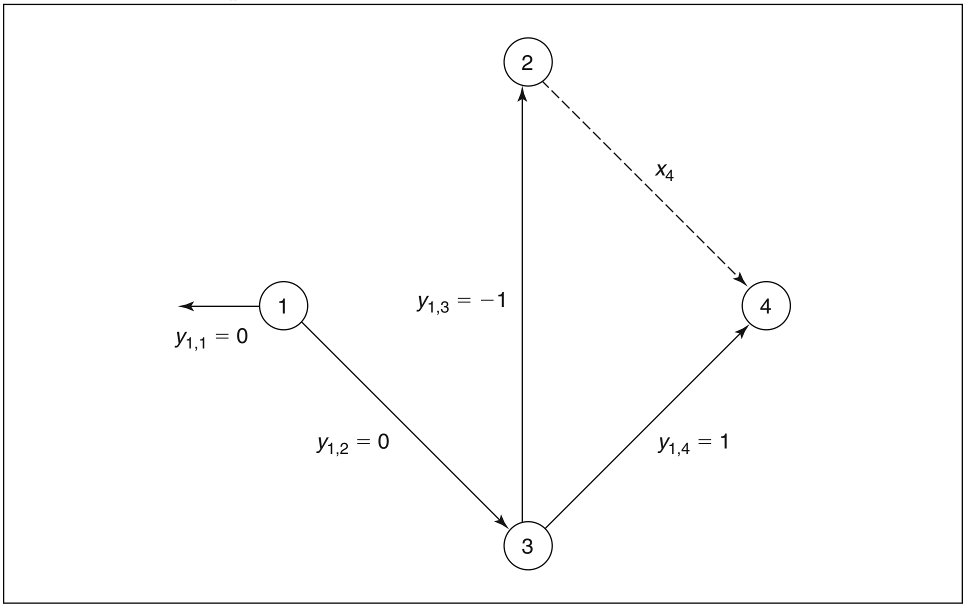
$$\mathbf{y}' = \begin{bmatrix} \mathbf{y}^1 \\ \mathbf{y}^2 \end{bmatrix} \quad \text{and} \quad \overline{\mathbf{A}}(k) = \begin{bmatrix} \mathbf{A}(k) \\ \mathbf{B}(k) \end{bmatrix}$$

If the entering column corresponds to arc[4], then

$$\begin{bmatrix} \mathbf{y}^1 \\ \mathbf{y}^2 \end{bmatrix} = \begin{bmatrix} \mathbf{G}^{-1} + \mathbf{G}^{-1}\mathbf{H}\mathbf{Q}^{-1}\mathbf{D}\mathbf{G}^{-1} & -\mathbf{G}^{-1}\mathbf{H}\mathbf{Q}^{-1} \\ -\mathbf{Q}^{-1}\mathbf{D}\mathbf{G}^{-1} & \mathbf{Q}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{A}(k) \\ \mathbf{B}(k) \end{bmatrix}$$
$$= \begin{bmatrix} \mathbf{G}^{-1}\{\mathbf{A}(k) + \mathbf{H}\mathbf{Q}^{-1}\mathbf{D}\mathbf{G}^{-1}\mathbf{A}(k) - \mathbf{H}\mathbf{Q}^{-1}(k)\} \\ \mathbf{Q}^{-1}\{\mathbf{B}(k) - \mathbf{D}\mathbf{G}^{-1}\mathbf{A}(k)\} \end{bmatrix} \tag{A4.16}$$

Step 1. Considering the entering variable x_4, perform the intermediate column update $\mathbf{y}_1 = \mathbf{G}^{-1}\mathbf{A}(4)$ using part of Equation A4.16. This calculation is shown in Figure A4.11. Here $P = \{2\ 3\ 4\}$, which when matched against the arrows at the arrival node, shows that $\mathbf{y}_1 = (0\ \ 0\ -1\ +1)^T$.

Figure A4.11 $\mathbf{y}_1$ CALCULATION IN RATIO TEST

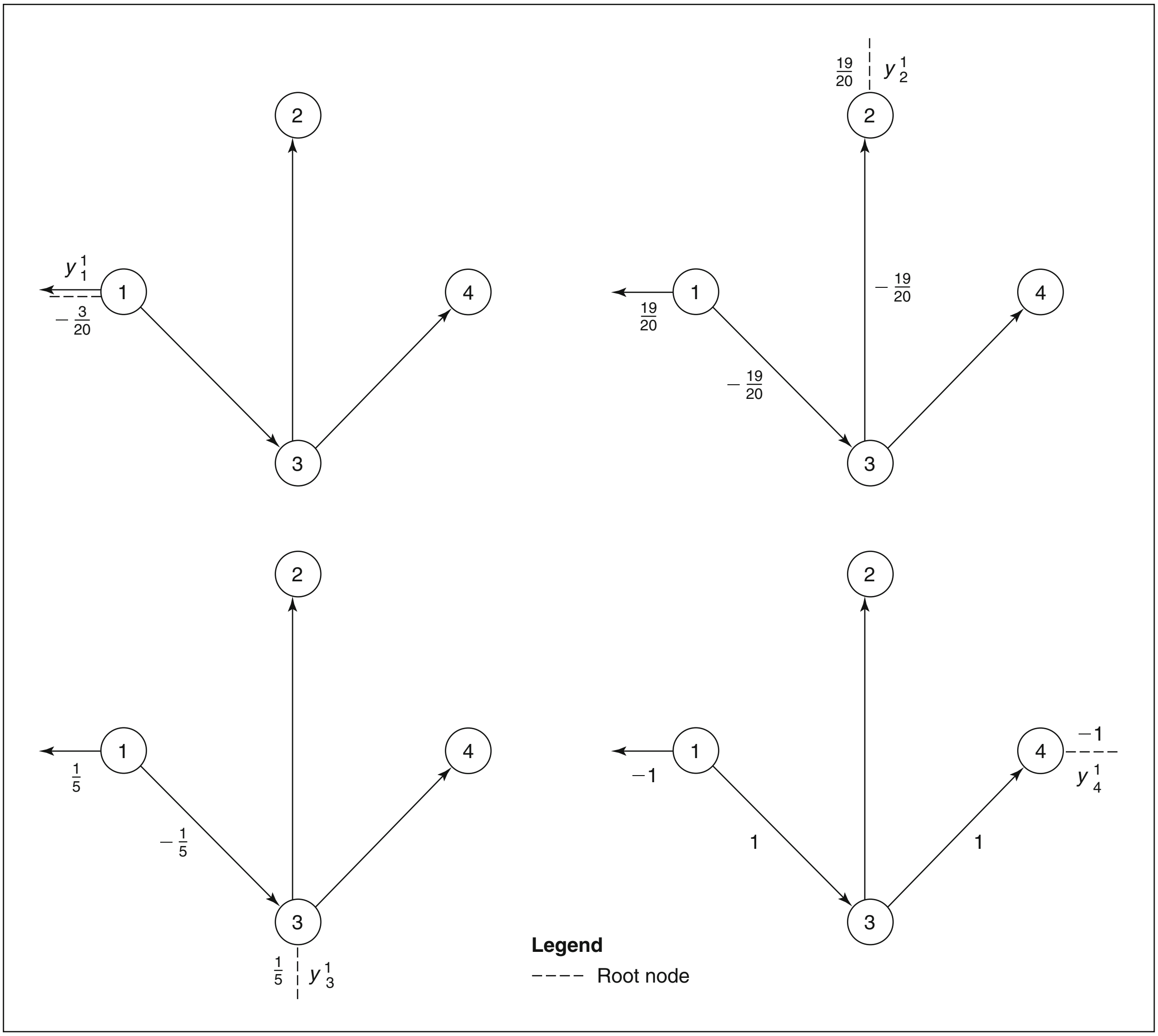


Step 2. From the first entry of the vector in Equation A4.16, define $\mathbf{y}_2 = \mathbf{A}(4) + \mathbf{HQ}^{-1}\mathbf{D}\mathbf{y}_1 - \mathbf{HQ}^{-1}\mathbf{B}(4)$

$$\mathbf{y}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ -1 & 1 \\ 0 & -1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1/12 & 1/15 \\ 0 & 1/5 \end{bmatrix} \begin{bmatrix} 0 & 0 & -2 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ -1 & 1 \\ 0 & -1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1/12 & 1/15 \\ 0 & 1/5 \end{bmatrix} \begin{bmatrix} 3 \\ 0 \end{bmatrix} = \begin{bmatrix} -3/20 \\ 19/20 \\ 1/5 \\ -1 \end{bmatrix} \tag{A4.17}$$

Step 3. Again, from the first entry of the vector in Equation A4.16, $\mathbf{y}^1 = \mathbf{G}^{-1}\mathbf{y}_2$, where the basis $\mathbf{G}$ is inverted by graphic means in Figure A4.12. In these figures, we show the computations of $\mathbf{y}_1^1$ by tracing the path from node 1 to node 1 in T_B, $\mathbf{y}_2^1$ by tracing from 2 to 1, $\mathbf{y}_3^1$ from 3 to 1, and $\mathbf{y}_4^1$. Notice that instead

Figure A4.12 CALCULATION OF $\mathbf{y}^1$



of from node 1 to all other nodes, we are starting from other nodes to node 1, since we have the system $\mathbf{G}^{-1}\mathbf{y}$ rather than $\pi^T\mathbf{G}^{-1}$ to solve.

From the graphs, or referencing the orientation sequences already contained in

$$\mathbf{G}^{-1} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -1 & -1 & -1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

we write

$$\mathbf{y}^1 = \begin{bmatrix} -3/20 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 19/20 \\ -19/20 \\ -19/20 \\ 0 \end{bmatrix} + \begin{bmatrix} 1/5 \\ -1/5 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -3/20 \\ -19/20 \\ 1 \end{bmatrix}$$

Step 4. According to Equation (A4.16), $\mathbf{y}^2 = \mathbf{Q}^{-1}[\mathbf{B}(4) - \mathbf{D}\mathbf{y}_1]$. Hence

$$\mathbf{y}^2 = \begin{bmatrix} 1/12 & 1/15 \\ 0 & 1/5 \end{bmatrix} \left\{ \begin{bmatrix} 3 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 & -2 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \end{bmatrix} \right\} = \begin{bmatrix} 3/20 \\ 1/5 \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{y}^1 \\ \mathbf{y}^2 \end{bmatrix} = (0\ \ -3/20\ \ -19/20\ \ 1\ 3/20\ \ 1/5)^T.$$

Similarly

$$\overline{\mathbf{B}}^{-1} \begin{bmatrix} \mathbf{b} \\ \mathbf{b'} \end{bmatrix} = (0\ \ 8\ \ 0\ \ 10\ \ 22) \qquad \text{(A4.18)}$$

For the column corresponding to the entering variable, we perform two types of ratio tests, since we have both a lower bound, 0, and upper bound, d_k, on the decision variables. The following tests correspond to the condition under which the basic variable drops to its lower bound or reaches its upper bound respectively:

$$\Delta_1 = \operatorname*{Min}_{1\le j\le m} \left\{ \frac{x_j^B - 0}{|y_j|}, \infty \right\} = \left\{ \frac{x_j^B}{|y_j|}, \infty \right\} = \operatorname{Min} \left\{ \frac{10}{1}, \frac{2}{\frac{3}{20}}, \frac{2}{\frac{1}{5}} \right\} = 10 \qquad \text{(A4.19)}$$

for positive entries in $\mathbf{y}'$, and

$$\Delta_2 = \operatorname*{Min}_{1\le j\le m} \left\{ \frac{d_j^B - x_j^B}{|y_j|}, \infty \right\} = \operatorname{Min} \left\{ \frac{18 - 8}{\frac{3}{20}}, \frac{1 - 0}{\frac{19}{20}} \right\} = \frac{20}{19} \qquad \text{(A4.20)}$$

for negative entries.

Considering the case when the entering variable x_k can reach its upper bound and hence be the restricting variable, the tolerable increase in resource engagement overall is

$$\Delta = \mathrm{Min}\{\Delta_1, \Delta_2, d_4\} = \mathrm{Min}\{10, 20/19, 12\} = 20/19 \tag{A4.21}$$

In this case, the leaving variable will be x_5 remembering the order of the variables in $\overline{\mathbf{B}}$ *and* $\overline{\mathbf{B}}^{-1}$ is x_7, x_2, x_5, x_6, x_1, and x_3.

This completes a simplex-on-a-graph iteration, consisting of one pricing operation and one ratio test for the sample problem. The network-with-side-constraint algorithm is used in locating satallite tracking stations in the "Facility Location" chapter under the "Generalized *p*-Median Problem" section in Chan (2005).

III. LAGRANGIAN RELAXATION

As can be seen from network with side constraints, for large-scale linear programs (LPs) or mixed integer programs (MIPs) with a special structure decomposition methods can be employed for computational efficiency. The central idea is to exploit the nice properties of the well-structured part of the mathematical program (such as a network matrix) and to set aside the more complicated part in the interim. Hence the term relaxation is sometimes used in the general decomposition procedure of this kind.

A. Illustration of Basic Concepts

A more general way to introduce decomposition is through Lagrange relaxation, which we will explain through an integer programming (IP) example (Fisher 1985)

$$\begin{array}{llllll} \text{Min } z_{IP} = & -16x_1 & -10x_2 & & -4x_4 & \\ \text{s.t.} & -8x_1 & -2x_2 & -x_3 & -4x_4 & \geq -10 \\ & -x_1 & -x_2 & & & \geq -1 \\ & & & -x_3 & -x_4 & \geq -1 \\ & & x_j = 0 \text{ or } 1 \text{ for all } j & & & \end{array} \tag{P}$$

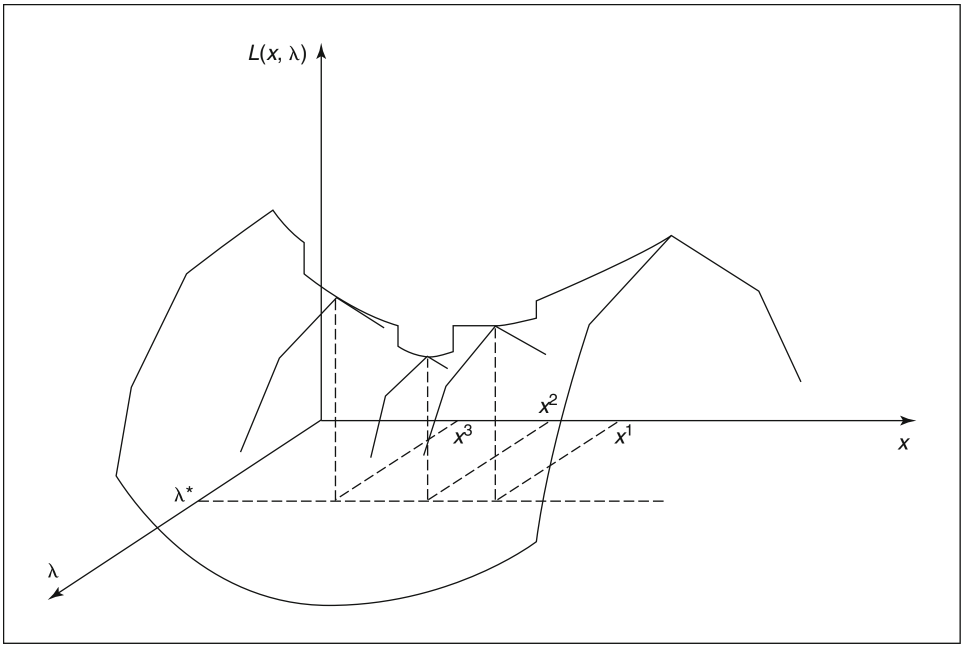
which has the form

$$\begin{aligned} &\text{Min } z_{IP} = \mathbf{c}^T\mathbf{x} \\ &\mathbf{A}^1\mathbf{x} \geq \mathbf{b}^1 \\ &\mathbf{A}^2\mathbf{x} \geq \mathbf{b}^2 \end{aligned} \tag{A4.22}$$

The first constraint is judged to be the complicated one and we form the relaxed Lagrangian by dualizing it:

$$\begin{array}{ll} z_{\mathrm{LR}}(\lambda) & = \mathrm{Min}\,[-16x - 10x_2 - 4x_4 + \lambda(-10 + 8x_1 + 2x_2 + x_3 + 4x_4)] \\ & = \mathrm{Min}\,[-x_1(16 - 8\lambda) - x_2(10 - 2\lambda) - \mathrm{x}_3(0 - \lambda) - x_4(4 - 4\lambda) - 10\lambda] \\ \text{s.t.} & -x_1 - x_2 \geq -1 \\ & -x_3 - x_4 \geq -1 \\ & x_j \in \{0, 1\} \quad \text{for all } j \end{array} \tag{LR($\boldsymbol{\lambda}$)}$$

Figure A4.13 MIN IN x AND MAX IN $\boldsymbol{\lambda}$ FOR A WEAK DUALITY



Here we have formed the Lagrangian relaxation problem

$$\begin{aligned} z_{\mathrm{LR}}(\boldsymbol{\lambda}) &= \mathrm{Min}_{\mathbf{x}}\,[\mathbf{c}^T\mathbf{x} + \boldsymbol{\lambda}^T(\mathbf{b}^1 - \mathbf{A}^1\mathbf{x})] \\ \text{s.t. } & \mathbf{A}^2\mathbf{x} \geq \mathbf{b}^2 \\ & \mathbf{x}_j = \{0, 1\} \quad \text{for all } j \end{aligned} \qquad (\mathrm{LR}(\boldsymbol{\lambda}))$$

Notice that a network-with-side-constraints model can be formulated as a Langrangian relaxation problem when $\mathbf{A}^2 = \mathbf{A}$ and $\mathbf{A}^1 = [\mathbf{B}\ \mathbf{C}]$.

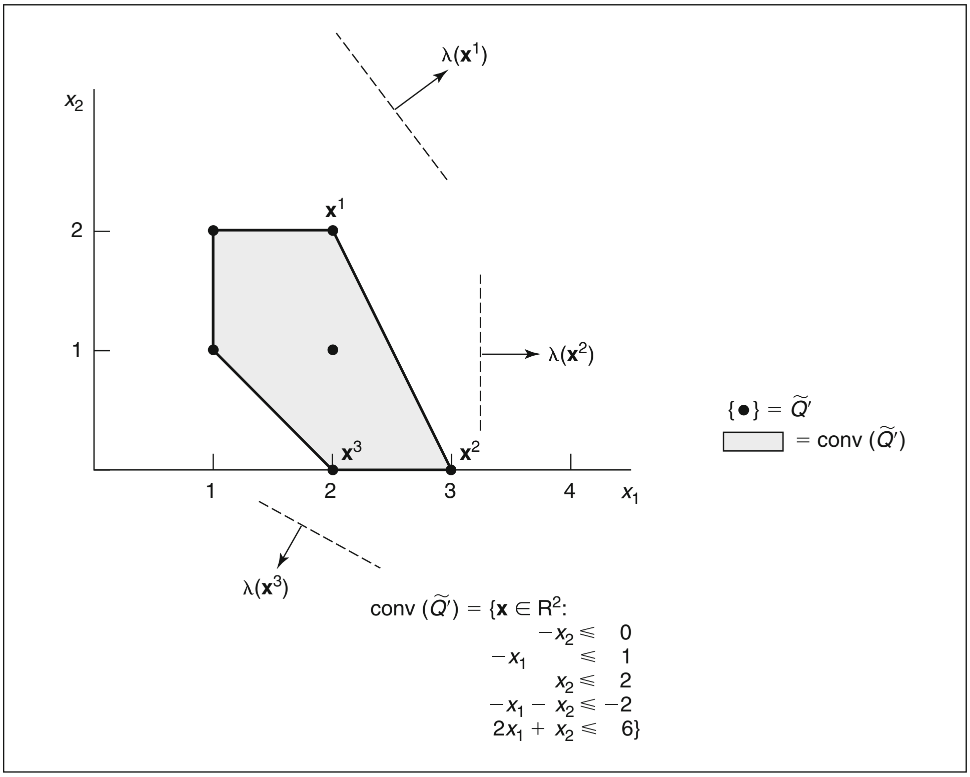
For the dual variable $\boldsymbol{\lambda}$ fixed at some non-negative value this problem is easy to solve (as a network flow problem for example), as shown in Figure A4.13 where the dual variable is fixed at its optimal value λ^*. The mathematical program reduces to minimizing over $\mathbf{x}$ (a discrete variable), yielding the optimal value at $\mathbf{x}^* = \mathbf{x}^2$. (Notice this includes the case of multiple optima.) For any other values of $\boldsymbol{\lambda}$, a weak duality results, which says that the resulting optimization over $\mathbf{x}$ will yield a z value smaller than before relaxation—some kind of a super optimum. The inequality $z_{\mathrm{LR}}(\lambda) \leq z_{IP}$ allows LR(λ) to be used in place of LP to provide lower bounds (cuts) in a branch and bound (B&B) algorithm for IP, where the bounds are usually tighter than LP relaxation.[5] The solution $\mathbf{x}^*$ is optimal to IP if there is a λ^* such that $z_{\mathrm{LR}}(\boldsymbol{\lambda}^*) = z_{\mathrm{IP}}$.

B. Underlying Theory

Restating the above in more formal terms:

$$z_{\mathrm{LR}}(\boldsymbol{\lambda}) = \mathrm{Min}_{\mathbf{x}}\,\{z(\boldsymbol{\lambda}, \mathbf{x}) : \mathbf{x} \in \mathrm{conv}(\tilde{Q}')\} \qquad (\mathrm{LR})$$

Figure A4.14 OPTIMIZING OVER A CONVEX HULL



where the convex hull conv($\widetilde{Q}'$) is formed from a convex combination of discrete points defined by $\mathbf{A}^2\mathbf{x} \geq \mathbf{b}^2$. In transitioning from the LR($\boldsymbol{\lambda}$) to the (LR) formulation, the convex hull has to be constructed from the polyhedron $\mathbf{A}^2\mathbf{x} \geq \mathbf{b}^2$ by trimming the "excess," thus exposing the discrete points. An example of this can be shown in Figure A4.14 below. It is interesting to contrast this with LP relaxation, in which $z_{\text{LP}} = \text{Min}_{\mathbf{x}}\{\mathbf{c}^T\mathbf{x}\text{:}\ \mathbf{x} \in S\}$ where $S = \{\mathbf{x} \in R_+^n\text{: } \mathbf{A}\mathbf{x} \geq \mathbf{b}\}$. Here R_+^n is the domain of continuous non-negative variables, rather than the discrete variables that are of real interest. We can now view the Lagrangian relaxation problem as minimization over a set of discrete points:

$$z_{\text{LR}}(\boldsymbol{\lambda}) = \underset{x_i \in \widetilde{Q}'}{\text{Min}}\ z(\boldsymbol{\lambda}, \mathbf{x}')$$

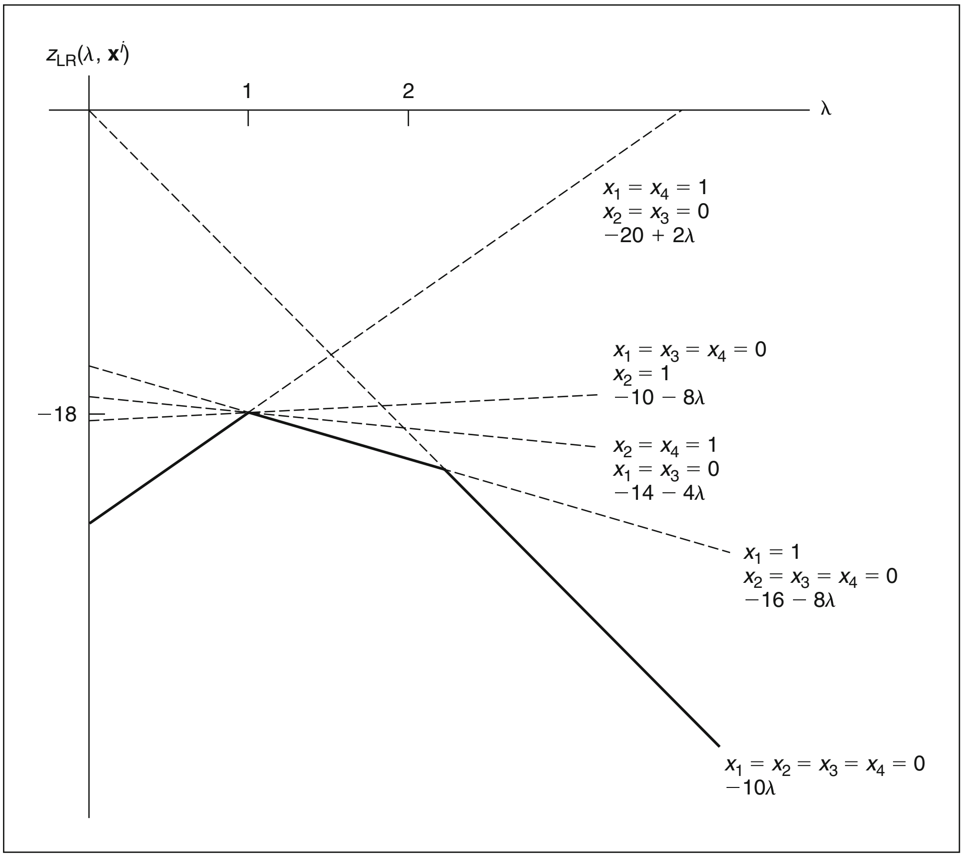
and to observe that for fixed $\mathbf{x}^i$, $z(\boldsymbol{\lambda}, \mathbf{x}^i) = \mathbf{c}^T\mathbf{x}^i + \boldsymbol{\lambda}(\mathbf{b}^1 - \mathbf{A}^1\mathbf{x}^i)$ is a function of $\boldsymbol{\lambda}$.

Ideally, $\boldsymbol{\lambda}$ should solve the Lagrangian dual problem z_{LD} in accordance with expression (LD) below, which provides the best choice of λ.

$$z_{\text{LD}} = \underset{\boldsymbol{\lambda} \leq 0}{\text{Max}}\ z_{\text{LR}}(\lambda) \qquad \text{(LD)}$$

z_{LD} is always linear in $\boldsymbol{\lambda}$. Based on the constraints of the relaxed program, a finite number of combinations for x_j are admissible, forming the feasible solutions $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^{|J|}$. These values and the resultant piecewise linear function are shown in Figure A4.15. Contrast this with LP relaxation in which $z_{\text{LP}}(\mathbf{u}) = \text{Max}_{\mathbf{u}}\{\mathbf{b}^T\mathbf{u}\text{:}\mathbf{u} \in P_{\text{D}}\}$ where

Figure A4.15 MAXIMIZATION IN DUAL VARIABLE



$P_D = \{\mathbf{u} \in R_+^m : \mathbf{A}^T\mathbf{u} \leq \mathbf{c}\}$ and $z_{LP}(\mathbf{u}) \leq z_{IP}$. Here, columns are appended to the dual of the LP corresponding to the imposition of integer values in the branching process. Until all variables are integerized, LP relations will yield super-optima. In Lagrangian relaxation, we are expressing LD formally as an LP with many constraints:

$$z_{LR}(\boldsymbol{\lambda} = \underset{\boldsymbol{\lambda} \geq \mathbf{0}}{\text{Max}} \{z' : z' \leq z(\boldsymbol{\lambda}, \mathbf{x}^i) \text{ for } i = 2, \ldots, |J|\} \qquad \text{(LD}'\text{)}$$

In other words:

$$\begin{array}{lrr} \text{Max } z' & & \\ \text{s.t.} & -20 + 2\lambda \geq z' & -10 - 8\lambda \geq z' \\ & -16 - 2\lambda \geq z' & -10\lambda \geq z' \\ & -14 - 4\lambda \geq z' & \lambda \geq 0 \end{array}$$

This problem is sketched out in Figure A4.15. Problem (LD′) makes it apparent that $z_{LR}(\boldsymbol{\lambda})$ is the lower envelope of a finite family of linear functions. The function $z_{LR}(\boldsymbol{\lambda})$ has all the nice properties, like continuity and concavity, that lend themselves to hill climbing algorithms (specifically subgradient optimization) (Ahuja et al. 1993; Nemhauser and Wolsey 1988).

C. Subgradient Optimization

It is appropriate at this juncture to explain the subgradient optimization algorithm (Fisher 1985). At differentiable points of Figure A4.15, the derivative of $z_{\mathrm{LR}}(\lambda)$ with respect to λ is given by $\mathbf{A}^1\mathbf{x} - \mathbf{b}^1$ or $8x_1 + 2x_2 + x_3 + 4x_4 - 10$ in our example, where $\mathbf{x}$ is an optimal solution to LR($\boldsymbol{\lambda}$). These facts also hold in general with the gradient of the $z_{\mathrm{LR}}(\boldsymbol{\lambda})$ function at differentiable points given by $\mathbf{A}^1\mathbf{x} - \mathbf{b}^1$, where $\mathbf{x}$ may not be optimal. This observation suggests it might be fruitful to apply a gradient search method to maximize $z_{LR}(\boldsymbol{\lambda})$ with some adaptation at the points where $z_{\mathrm{LR}}(\boldsymbol{\lambda})$ is nondifferentiable. The subgradient method chooses arbitrarily from the set of alternative optimal Lagrangian solutions $\mathbf{x}^i$ at these nondifferentiable points and use the vector $\mathbf{A}^1\mathbf{x} - \mathbf{b}^1$ for this solution as though it were the gradient of LD′. The result is a procedure that determines a sequence of values for $\boldsymbol{\lambda}$ by beginning at an initial point $\boldsymbol{\lambda}^0$ (such as zero) and applying the following formula. We illustrate this problem for the case where $\boldsymbol{\lambda}$ is scalar:

$$\lambda^{k+1} = \mathrm{Max}\ [0, \lambda^k - t_k(\mathbf{b}^1 - \mathbf{A}^1\mathbf{x}^k] \qquad \text{(A4.23)}$$

In this formula, t_k is a scalar step size and $\mathbf{x}^k$ is an optimal solution to LR($\boldsymbol{\lambda}^k$), the Lagrangian problem with dual variables set to λ^k. Equation A4.23 can be thought of as a generalization of the method of steepest ascent in nonlinear programming when the objective function is piecewise linear (See Section III-F in Chapter 4). The choice is between staying put or moving along a gradient, whichever is better. Even though we have illustrated subgradient optimization only through the scalar example, multiple dimension generalization of Equation A4.23 can be found in the formalization below and in Nemhauser and Wolsey (1988) and Reeves (1993).

The nondifferentiability also requires some variation in the way the step size is normally set in a gradient method. A formula for t_k that has proven effective in practice is

$$t_k = \frac{\tau_k(z_{\mathrm{LR}}(\lambda^k) - z*_{\mathrm{IP}})}{\sum_{i=1}^{m^1} (b_i^1 - \sum_{j=1}^{n} a_{ij}^1 x_j^k)^2} = \frac{\tau_k(z_{\mathrm{LE}}(\lambda^k) - z*_{\mathrm{IP}})}{\| \mathbf{b}^1 - \mathbf{A}\mathbf{x}^k \|} \qquad \text{(A4.24)}$$

In this formula, z^*_{IP} is the objective function value of the best known feasible solution to the original problem P and τ_k is a user defined scalar chosen between 0 and 2. It is assumed here that there are m^1 complicated constraints in $\mathbf{A}^1\mathbf{x} \geq \mathbf{b}^1$. Notice Equation A4.24 measures the difference between z^* and the current Lagrangian objective against the Euclidean norm (or the l_2-norm as described in Chapter 5 of this book and the "Measuring Spatial Separation" chapter in Chan (2005). When the difference is large relative to the norm, a larger step size is taken and vice versa. Frequently, the sequence τ_k is determined by starting with $\tau_k = 2$ and reducing τ_k by a factor of two whenever LR($\boldsymbol{\lambda}^k$) has failed to increase in a specified number of iterations. The feasible value z^* initially can be set to 0 and then updated using the solutions that are obtained on those iterations, in which the Lagrangian problem solution turns out to be feasible in the original problem P. Unless we obtained a λ^k for which

$\mathrm{LR}(\boldsymbol{\lambda}^k) = z^*_{\mathrm{IP}}$, there is no way of proving optimality in the subgradient method. To resolve this difficulty, the algorithm is usually terminated upon reaching a specified iteration limit.

Example
Here is an example of the subgradient method illustrating the judicious choice of step sizes:

k	*Dual variable* λ_k	*Step size* t_k
0	0	1
1	Max[0, 0 − (1)(−2)] = 2	1/2
2	Max[0, 2 − (1/2)(8)] = 2	1/4
3	Max[0, 0 − (1/4)(−2)] = 1/2	1/8
4	Max[0, 1/2 − (1/8)(−2)] = 3/4	1/16
5	Max[0, 3/4 − (1/16)(−2)] = 7/8	1/32
6	Max[0, 7/8 − (1/32)(−2)] = 15/16	etc.

As can be seen, the algorithm converges nicely to the optimal value of $\lambda = 1$. ■

Subgradient Optimization Algorithm

The subgradient algorithm can now be applied to the Langrangian relaxation problem as follows:

Step 1: Solve the Lagrangian relaxation problem $\mathrm{LR}(\boldsymbol{\lambda}^k)$ to obtain the optimal $\mathbf{x}^k$.
Step 2: Evaluate the subgradient $\mathbf{g}(\boldsymbol{\lambda}^k) = \mathbf{b}^1 - \mathbf{A}^1\mathbf{x}^k$. If $\mathbf{g}(\boldsymbol{\lambda}^k) = \mathbf{0}$, stop; $(\boldsymbol{\lambda}^k, \mathbf{x}^k)$ is an optimal solution.
Step 3: Let $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + t_k\mathbf{g}(\boldsymbol{\lambda}^k)$, which is a m^1-entry vector generalization of Equation A4.23. Increment counter $k + 1 \rightarrow k$, and go to step 1.

From Figure A4.15, it is easy to see that $\lambda = 1$ maximizes $z_{\mathrm{LR}}(\lambda)$. Thus the lower bound is $z_{\mathrm{LR}}(1) = -18$ and a corresponding feasible solution of $z_{\mathrm{IP}} = -16$ by inspection, namely one of three feasible solutions[6]: $\mathbf{x} = (1, 0, 0, 0)$, $(0, 1, 0, 0)$, or $(0, 1, 0, 1)$. The solution $(1, 0, 0, 0)$ yields $z_{\mathrm{IP}}(1, 0, 0, 0) = -16$. Formally, the lower bound -18 should now be used in B&B to arrive at the optimal solution $\mathbf{x}^*$. In other words, by taking a convex combination of points in $\widetilde{Q}'$, we obtain a point $\boldsymbol{x}^*$ in $\mathrm{conv}(\widetilde{Q}')$ satisfying the complicating constraint, for which $\mathbf{c}^T\mathbf{x}^* = z_{\mathrm{LD}}$. This shows that for the example we obtain $z_{\mathrm{LD}} = \mathrm{Min}\{\mathbf{c}^T\mathbf{x}{:}\mathbf{A}^1\mathbf{x} \geq \mathbf{b}^1,\ \mathbf{x} \in \mathrm{conv}(\widetilde{Q}')\}$. The major result is as follows: The primal LP problem of finding a convex combination of points in $\widetilde{Q}'$ that also satisfies the complicating constraint $\mathbf{A}^1\mathbf{x} \geq \mathbf{b}^1$ or $-8x_1 - 2x_2 - x_3 - 4x_4 \geq -10$ is dual to the Lagrangian dual, or the solution is optimal.

D. Branch-and-Bound (B&B) Solution

While we showed above how the optimum can be obtained, this is not simple in general. Oftentimes, we need to resort to a tree search (B&B) procedure to resolve the problem (as explained in Section III-B of Chapter 4). We use the traditional B&B procedure in which a tree of solution alternatives is constructed with certain variables fixed to specified values at each node of the tree, representing a proposal $\mathbf{x}^{i_k}$.

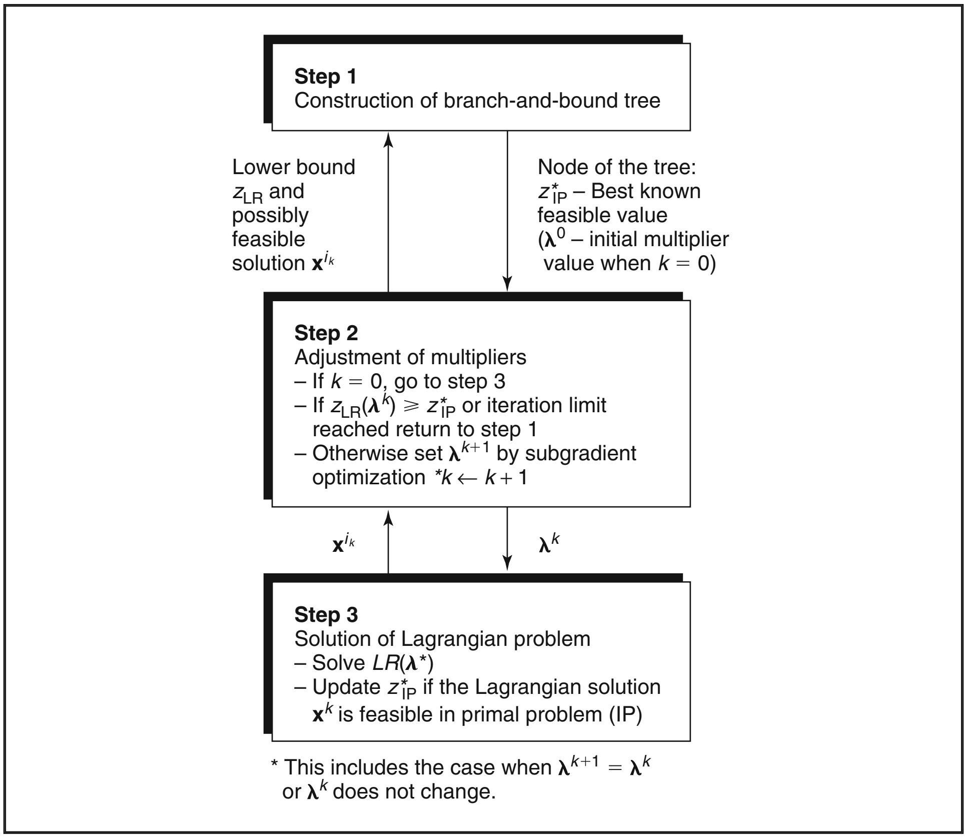
Shown below is a tabular display of the various contending solutions at $\lambda = 1$ in Figure A4.15. It serves to illustrate a B&B tree that can lead toward the optimal solution via pruning rules such as infeasibility, dominance, and incumbency[7]:

λ	Lagrangian solution					z_{IP}
	x_1	x_2	x_3	x_4	$z_{LR}(\lambda)$	
1	1	0	0	0	−18	−16*
1	1	0	0	1	−18	Infeasible
1	0	1	0	0	−18	−10
1	0	1	0	1	−18	−14

For example, a B&B tree will be pruned at the infeasible solution node as represented by the second line of the table. Similarly, the solutions on the third and fourth lines are dominated by the solution of −16 in the first line. All things said and done, here the solution at the first line is the optimum, yielding $\mathbf{x}^* = (1\ 0\ 0\ 0)$ and $z^*_{IP} = 16$.

Insightful as this example may be, left unexplained is the procedure to generate the $\mathbf{x}^i$, from which the constraints of (LD′) can be generated. For a

Figure A4.16 GENERIC LAGRANGIAN-RELAXATION ALGORITHM


Step 1
Construction of branch-and-bound tree
Lower bound z_{LR} and possibly feasible solution $\mathbf{x}^{i_k}$
Node of the tree:
z^*_{IP} – Best known feasible value
($\boldsymbol{\lambda}^0$ – initial multiplier value when $k = 0$)
Step 2
Adjustment of multipliers
– If $k = 0$, go to step 3
– If $z_{LR}(\boldsymbol{\lambda}^k) \geqslant z^*_{IP}$ or iteration limit reached return to step 1
– Otherwise set $\boldsymbol{\lambda}^{k+1}$ by subgradient optimization $^*k \leftarrow k+1$
$\mathbf{x}^{i_k}$
$\boldsymbol{\lambda}^k$
Step 3
Solution of Lagrangian problem
– Solve $LR(\boldsymbol{\lambda}^*)$
– Update z^*_{IP} if the Lagrangian solution $\mathbf{x}^k$ is feasible in primal problem (IP)
* This includes the case when $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k$ or $\boldsymbol{\lambda}^k$ does not change.

SOURCE: Fisher (1985). Reprinted with permission.

totally unimodular matrix, such as the one shown in $\mathbf{A}^2$, the $\mathbf{x}^i$s are discernable. However, this is not the case in general, when the extreme points of the polyhedron $\mathbf{A}^2\mathbf{x} \geq \mathbf{b}^2$ are not integer valued. Suppose we start out with a couple of integer points $\mathbf{x}^{i0}$ on an integer convex-hull $\widetilde{Q}'$, from which $z_{\mathrm{LD}}(\boldsymbol{\lambda})$ can be maximized at $\boldsymbol{\lambda}^0$. Then for $\boldsymbol{\lambda}^{0\prime} \neq \boldsymbol{\lambda}^0$ we solve for $z_{\mathrm{LR}}(\boldsymbol{\lambda}^{0\prime}, \mathbf{x})$ according to (LR) for additional $\mathbf{x}^{ij}$ with which (LD′) can be solved again, often with the help of a B&B tree. The process then gets repeated again for the $\boldsymbol{\lambda}^1$ so obtained, until two subsequent iterations yield the same $z_{\mathrm{LR}}(\boldsymbol{\lambda}^*, \mathbf{x}^*)$. The generic Lagrangian-relaxation algorithm is illustrated in Figure A4.16. This figure shows the complete Lagrangian relaxation algorithm consisting of three major steps. The first step is the standard B&B process in which a tree of solution alternatives $\{\mathbf{x}^k\}$ is generated with certain variables fixed to specified values at each node of the tree, namely the $\boldsymbol{\lambda}^k$ values. These specified values are passed from block 1 to block 2 together with z^*_{IP}, the objective function value of the currently best-known feasible solution. In the initial step, we set $k = 0$ and the value of the starting multipliers $\boldsymbol{\lambda}^0$ (normally to $\mathbf{0}$).

We iterate between blocks 2 and 3, adjusting the multipliers $\boldsymbol{\lambda}^k$ with the vector generalization of the subgradient update and obtaining a new Lagrangian solution $\mathbf{x}^k$ respectively. This process continues until we either reach an interation limit or discover an upper bound that is less than or equal to the current best-known feasible solution z^*_{IP}. At this point, we pass back to block 1 the best upper-bound together with any feasible solution z^*_{IP} that may have been obtained as a result of solving the Lagrangian problem LR($\boldsymbol{\lambda}^k$).

According to Fisher (1985), it is not uncommon in large-scale applications to terminate the process depicted in Figure A4.16 before the B&B tree has been explored sufficiently to prove optimality. Beasley shared many of his computational experiences in Reeves (1993). In this case, the Lagrangian algorithm is really a heuristic—similar to LP relaxation—with some nice properties, such as the maximum amount by which the heuristic solution z^*_{IP} deviates from optimality. Related discussions on Lagrangian relaxation can be found in Lubbecke and Desrosiers (2005), Chapter 4, Section V in this book and in the "Facility Location" chapter in Chan (2005) under "Median Location Problems."

IV. BENDERS' DECOMPOSITION

Lagrangian relaxation takes care of complicating constraints by incorporating them in the objective function. Here we consider the allied problem of complicating variables. Suppose we have the following mixed integer program (MIP)

$$\begin{aligned} z &= \mathrm{Max}\ (\mathbf{g}^T\mathbf{y} + \mathbf{c}^T\mathbf{x}) \\ \text{s.t.} &\quad \mathbf{By} + \mathbf{Ax} \leq \mathbf{b} \end{aligned}$$

where $\mathbf{y}$ are non-negative discrete variables of dimension n ($\mathbf{y} \in Y'' \subseteq Z''_+$), *and* $\mathbf{x}$ are continuous non-negative variables of dimension p ($\mathbf{x} \in R^p_+$). We think of the discrete variables $\mathbf{y}$ as complicating variables to what would otherwise be a linear program (LP), or we can view the continuous variables $\mathbf{x}$ as complicating variables to what would have been a pure integer program. Instead of Lagrangian relaxation, an allied procedure called Benders' decomposition is employed to solve this MIP.

A. Example

Let us illustrate with a numerical example (Nemhauser and Wolsey 1988).

$$\begin{aligned}
\text{Max}\quad & 5y_1 - 2y_2 + 9y_3 + 2x_1 - 3x_2 + 4x_3 \\
\text{s.t.}\quad & 5y_1 - 3y_2 + 7y_3 + 2x_1 + 3x_2 + 6x_3 \leq -2 \\
& 4y_1 + 2y_2 + 4y_3 + 3x_1 - x_2 + 3x_3 \leq 10 \\
& y_j \leq 5 \text{ for } j = 1, 2, 3 \\
& \mathbf{y} \in Z_+^3, \mathbf{x} \in R_+^3
\end{aligned} \tag{A4.25}$$

Here

$$Y = \{\mathbf{y} \in Z_+^3 : y_j \leq 5 \text{ for } j = 1, 2, 3\}$$

As a first step, we suppose that the integer variables **y** have been fixed, in other words, projecting on **y**. The resulting LP is:

$$z_{\text{LP}}(\mathbf{x}) = \text{Max}\{\mathbf{c}^T\mathbf{x}\text{:}\mathbf{Ax} \leq \mathbf{b} - \mathbf{By}, \mathbf{x} \in \text{R}_+^p\} \tag{A4.26}$$

and its dual is

$$\text{Min}\{(\mathbf{b} - \mathbf{By})\, \boldsymbol{\lambda} : \text{A}\boldsymbol{\lambda} \geq \boldsymbol{c}, \boldsymbol{\lambda} \in R_+^m\} \tag{A4.27}$$

which forms a subproblem. For our current example, we have the dual polyhedron $\{\mathbf{A}\boldsymbol{\lambda} \geq \mathbf{c}, \boldsymbol{\lambda} \in R_+^2\}$ of dimension 2 or

$$\begin{aligned}
2\lambda_1 + 3\lambda_2 &\geq 2 \\
3\lambda_1 - \lambda_2 &\geq -3 \\
6\lambda_1 + 3\lambda_2 &\geq 4 \\
\lambda \in \text{R}_+^2 &
\end{aligned}$$

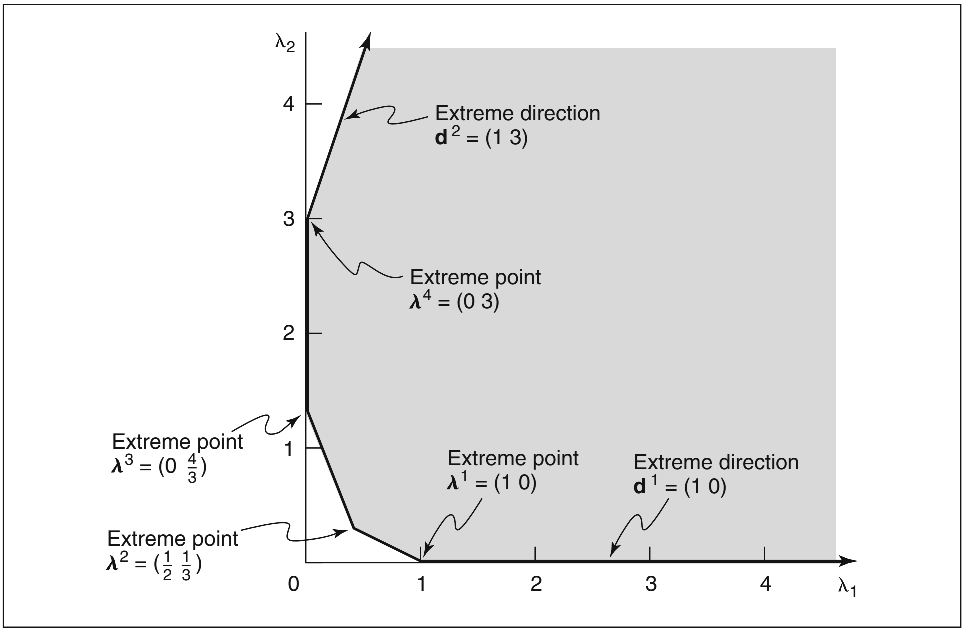
This polyhedron is sketched out in Figure A4.17, where the extreme points and extreme directions are shown. A bounded optimal solution can be represented by these extreme points and directions.

Let $\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2, \ldots, \boldsymbol{\lambda}^{|K|}$ be the extreme points and $\mathbf{d}^1, \mathbf{d}^2, \ldots, \mathbf{d}^{|J|}$ be the extreme directions of the dual polyhedron D''. Then any point $\boldsymbol{\lambda}$ in D'' can be represented by the extreme points and directions as we explained in Section I-B:

$$\begin{aligned}
\boldsymbol{\lambda} &= \sum_{k \in K} w_k \boldsymbol{\lambda}^k + \sum_{j \in J} \mu_j \mathbf{d}^j \\
\sum_{k \in K} w_k &= 1; \quad w_k, \mu_j \geq 0 \quad k \in K, j \in J
\end{aligned} \tag{A4.28}$$

If $z = \text{Max}_{\mathbf{x},\mathbf{y}}\{\mathbf{g}^T\mathbf{y} + \mathbf{c}^T\mathbf{x})$, then $z = \text{Max}_{\mathbf{y}}\{\mathbf{g}^T\mathbf{y} + \text{Min}\{\mathbf{b} - \mathbf{By})\boldsymbol{\lambda}\}$ for each $\boldsymbol{\lambda} \in D''$, *or* $z = \text{Max}_{\mathbf{y}}\{\mathbf{g}^T\mathbf{y} + \text{Min}_{\mathbf{j}\in\mathbf{J}}(\mathbf{b} - \mathbf{By})\boldsymbol{\lambda}^j\}$.· Here λ^j includes both extreme points and extreme directions. But if $(\mathbf{b} - \mathbf{By})\mathbf{d}^j < 0$ for some j, we can choose μ_j large enough so that

Figure A4.17 DUAL POLYHEDRON FOR BENDERS′ EXAMPLE



$$z = \mathrm{Max}_{\mathbf{y}}\{\mathbf{g}^T\mathbf{y} + \mathrm{Min}_{\boldsymbol{\lambda}\in D}\{(\mathbf{b} - \mathbf{B}\mathbf{y})\boldsymbol{\lambda}\}$$

becomes infeasible. Hence we must impose the additional constraints $(\mathbf{b} - \mathbf{B}\mathbf{y})\mathbf{d}^j > 0$. [For those $(\mathbf{b} - \mathbf{B}\mathbf{y})\mathbf{d}^j > 0$, naturally we would gravitate toward the extreme points and move away from the extreme directions by setting the appropriate μ_{j} to zero.] The MIP can now be rewritten as

$$\begin{gathered} z = \mathrm{Max}_{\mathbf{y}}\{\mathbf{g}^T\mathbf{y} + \underset{k\in K}{\mathrm{Min}}(\mathbf{b} - \mathbf{B}\mathbf{y})\boldsymbol{\lambda}^k\} \\ (\mathbf{b}-\mathbf{B}\mathbf{y})\mathbf{d}^j \geq 0 \quad \text{for } j \in J \\ \mathbf{y} \in \mathbf{Y}'' \end{gathered} \tag{A4.29}$$

and the problem can be reformulated as the master problem

$$\begin{gathered} z = \mathrm{Max}\ z' \\ s.t.\ \ z' \leq \mathbf{g}^T\mathbf{y} + (\mathbf{b} - \mathbf{B}\mathbf{y})\boldsymbol{\lambda}^k \quad \text{for } k \in K \\ (\mathbf{b} - \mathbf{B}\mathbf{y})\mathbf{d}^j \geq 0 \quad \text{for } j \in J \\ \mathbf{y} \in \mathbf{Y}'' \end{gathered} \tag{A4.30}$$

In our case for the finite number extreme points $\boldsymbol{\lambda}$ and extreme directions $\mathbf{d}$, the complete master problem (which is an all-integer program) looks like

$$\begin{aligned} z = {} & \mathrm{Max}\ z' \\ s.t.\ \ & z' \leq 5y_1 - 2y_2 + 9y_3 + \quad (-2 - 5y_1 + 3y_2 - 7y_3 \\ & z' \leq 5y_1 - 2y_2 + 9y_3 + {}^1\!/_2(-2 - 5y_1 + 3y_2 - 7y_3) + {}^1\!/_3(10 - 4y_1 - 2y_2 - 4y_3) \\ & z' \leq 5y_1 - 2y_2 + 9y_3 \qquad\qquad\qquad\qquad\qquad\quad + {}^4\!/_3(10 - 4y_1 - 2y_2 - 4y_3) \end{aligned}$$

$$\begin{array}{llll} z' \leq 5y_1 - 2y_2 + 9y_3 & & + \; 3(10 - 4y_1 - 2y_2 - 4y_3) & \\ & -2 - 5y_1 + 3y_2 - 7y_3 & & \geq 0 \\ & -2 - 5y_1 + 3y_2 - 7y_3 & + \; 3(10 - 4y_1 - 2y_2 - 4y_3 & \geq 0 \\ & y_j \leq 5 \text{ for } j = 1, 2, 3 & & \\ & \quad \mathbf{y} \in Z_{+}^{3} & & \end{array}$$

It can be verified that an optimal solution is $\mathbf{y} = (0\ 3\ 1)^T$ and $z' = 3$. Using this information, it can be easily verified that $\mathbf{x} = (1\,0\,0\,0)^T$ is part of the optimal solution to the example. This can be seen by substituting $\mathbf{y} = (0\ 3\ 1)^T$ into Equation A4.25 and solve the resulting LP in $\mathbf{x}$ as represented by Equation A4.26.

B. Convergence

In practice, there are an enormous number of constraints in the above master problem. A natural approach is to consider relaxations obtained by generating only those constraints corresponding to a small number of extreme points $k = 1, 2, \ldots, k'$ and extreme directions, $j = 1, 2, \ldots, j'$. We call these the relaxed master problems, yielding an optimal solution $(z', \mathbf{y}'')$, which is an upper bound on z. The solution $(z', \mathbf{y}'')$ is optimal if and only if it is feasible to all constraints in the master problem. In other words, we wish to check the subproblem shown in Equation A4.27 where $\mathbf{y}$ is obtained from the optimal solution of the relaxed master problem [Equation A4.30]. Given a finite number of extreme points $k = 1, \ldots, |K|$ and a finite number of extreme directions $j = 1, \ldots, |J|$, an optimal solution $(z^*, \mathbf{y}^*)$ exists. Let $\boldsymbol{\lambda}'^*$ be an optimal extreme point corresponding to an iteration where $k' + 1$ extreme points and $j' + 1$ extreme directions have been generated. If the optimum is obtained, $z^* = \mathbf{g}^T\mathbf{y}^* + (\mathbf{b} - \mathbf{B}\mathbf{y}^*)\boldsymbol{\lambda}'^*$. Then $z' \leq \mathbf{g}^T\mathbf{y}^* + (\mathbf{b} - \mathbf{B}\mathbf{y}^*)\boldsymbol{\lambda}'^*$ for all k and j. Otherwise, generate additional extreme points and directions as necessary in accordance with the relaxed master problem. Notice the iterations are schematically represented in Figure A4.15 if we read $z_{\mathrm{LR}}(\boldsymbol{\lambda}, \mathbf{x}^i)$ as $z(\boldsymbol{\lambda}, \mathbf{y}^k)$ according to Equation A4.30, even though the figure was drawn initially for Lagrangian relaxation.

An example illustrating the Benders' convergence process is found in the "Measuring Spatial Separation" chapter in Chan (2005) under the "Scheduling Restrictions" subsection. This is supplemented by a homework problem in which each step of the algorithm is spelled out in detail.

C. Extension

The discussion above can easily be extended to the case where the linear term **gy** in the objective function is a nonlinear function $f(\mathbf{y})$. In the above algorithm, simply replace **gy** with $f(\mathbf{y})$ throughout. The only difference is in the way we solve the master problem, which may involve a nonlinear solver. Nonlinear integer programming is not an easy task, however. In principle, the Benders' scheme can still be applied, taking full advantage of the part of the mathematical program which can be solved as a subproblem by LP.

Benders' decomposition is sometimes referred to as a resource-directive decomposition method. It starts with an initial solution consisting of, for instance, the discrete variable **y** and dual variable and adjusts the common resource available **b** by fixing the next **y** through master problem (30) and the corresponding (and hence **x**-variable) values through the dual subproblem (27). As suggested by Figure A4.15, this resource-directive decomposition scheme can be viewed as an alternative approach to the subgradient search used in Lagrangian relaxation.

Benders' decomposition is also a competing approach for solving the multicommodity flow problem, formerly formulated in Section II-A as a network with side constraints. In this regard, the common flow-capacity linking constraint is **By** and the commodity-flows are modeled by **Ax**. The Benders' approach decomposes the problem into a separate single-commodity flow problem for each commodity by allocating the scarce bundle capacities to the various commodities. Finding the optimal allocation (in other words, the one that gives the overall lowest cost in this case) is an optimization problem with a simple constraint structure and a (complicated) convex cost objective function. Using sensitivity information about the single commodity subproblems, however, we can generate subgradient information about the resource-allocation cost function and solve the allocation problem by a version of the subgradient optimization technique. Interested readers are referred to Ahuja et al. (1993) for further details. Benders' decomposition is an important and viable technique in solving location-routing models, as illustrated in the "Generalized Benders' Decomposition" subsection in the "Simultaneous Location-and-Routing Models" chapter in Chan (2005).

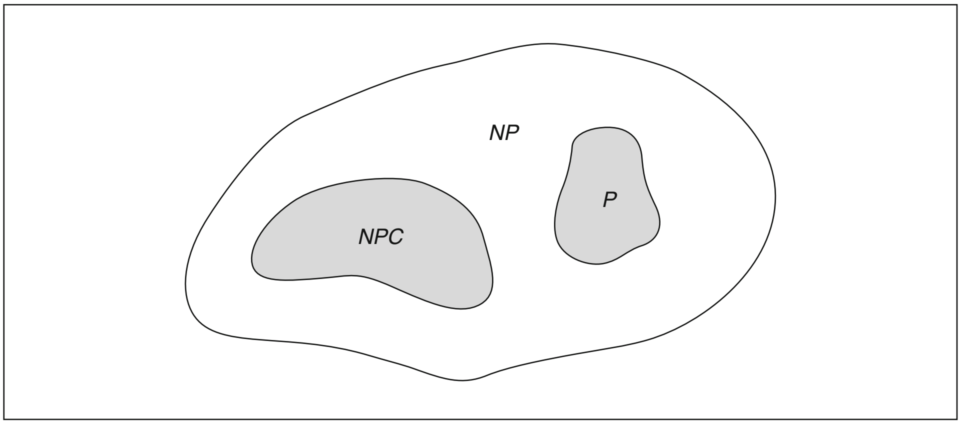
V. ALGORITHMS AND COMPLEXITY

Throughout this book, particularly in the discussions in this appendix, we are concerned with the efficiency of solution algorithms, which led to network-with- side-restraints and Benders' decomposition. The theory of computational complexity yields insights into how difficult a problem may be to solve and hence how much computational savings are obtainable from more efficient algorithms. For example, we may be able to show that in the order of $O(l^k)$ time, for some fixed k and data-input length l, an optimal solution is obtained. In this section, we wish to define some commonly used terms and to fix some basic notions.

Class *P* problems: Most efficient min-path algorithms, for example, are polynomial P in execution time. As shown in the literature, the complexity is $O(m)$ for a path, $O(m^2)$ for a tree and $O(m^3)$ for a point-to-point computation, where m is the number of nodes in a network. The polynomial order makes min-cost-flow network algorithms an attractive alternative to more computationally demanding methods such as regular simplex, as illustrated in Section II-B.

***NP* problems:** In contrast, simplex LP is a non-deterministic poly-nomial (NP) problem. In the simplex algorithm, the number of elementary steps required to solve the $m \times n$ LP is $O(mn)$ arithmetic operations for each pivot iteration, since it can be viewed as a matrix vector multiplication. The simplex algorithm can, at worst, visit all basic feasible solutions, and there are at most $\binom{m+n}{m}$ basic feasible solutions, requiring therefore $\binom{m+n}{m}$ pivots. All together, simplex is an $O(mn\binom{m+n}{m})$ algorithm. Notice this makes the simplex a much less efficient algorithm than network flow. The relationship between class P and NP is illustrated in Figure A4.18, in that polynomial problems are a subset, and a special case, of non-deterministic problems.

Figure A4.18 TYPES OF COMPLEXITY



SOURCE: Nemhauser & Wolsey (1988). Reprinted with permission.

***NP*-completeness:** An *NP*-complete (*NPC*) problem is a computational problem that is as hard as any reasonable problem; specifically, an *NPC* problem is characterized by:

1. No *NPC* problem can be solved by any known polynomial algorithm.
2. If there is a polynomial algorithm for any *NPC* problem, then there are polynomial algorithms for all *NPC* problems.

Figure A4.18 shows the class of *NP* and the two subsets *P* and *NPC*, which are disjoint unless $P = NP$, which put us in the category of item 2 above. If $P \neq NP$, it can be shown that $P \cup NPC \neq NP$.

Example
The (symmetric) traveling salesman problem (TSP) seeks to find the lowest cost tour among m nodes and return home.[8] There are $(m - 1)!/2$ possible tours in a network, where a tour (or Hamilton circuit) is a path traversing each node in the network exactly once. There is a solution to TSP if and only if a Hamilton circuit exists. TSP is an *NPC* problem (even though the Hamilton circuit problem may be a *P* problem). Finding the TSP tour among the 50 state capitals in the United States, for instance, could require many billions of years, with the fastest computer available. ■

***NP*-hard problems:** One problem polynomially reduces to another if a polynomially bounded number of calls to an algorithm for the second will always solve the first. Sometimes we may be able to show that all problems in *NP* polynomially reduce to some problem α''. But we are unable to argue that $\alpha'' \in NP$. So α'' does not qualify to be called *NP*-complete. Yet undoubtedly α'' is as hard as any problem in *NP*, and hence most probably intractable. It is for these problems that we have reserved the term *NP*-hard. A polynomial algorithm for an *NP*-hard problem implies $P = NP$. Problems that are both *NP*-hard and member of the class *NP* are called *NPC* (Figure A4.18.). It has turned out that the family of *NP*-hard problems is amazingly rich, including 0-1 integer/mixed integer programming (polynomial backtracking)[9].

To date, complexity theory is still evolving and may continue to do so for quite some time. With or without complexity theory, however, practitioners will continue to be confronted by significant discrete problems in all sorts of management and engineering settings. To date, the main value of the theory for practitioners has been to provide a theoretical base that confirms long held suspicions. Most importantly, it allows us to gauge the worst-case scenarios of efficiency independent of computer hardware and software—both of which are evolving too rapidly to allow for meaningful comparisons across diverse computational platforms. One must note that the worst-case behavior of an algorithm might be markedly different from its behavior in practice. Indeed, several *NP* and *NPC* algorithms can be solved very efficiently in practice. An example is the simplex algorithm, which has undergone generations of streamlining to make it competitive with polynomial LP solvers. However, in the words of Ahuja et al. (1993), we can safely say *NPC* problems sometimes do not have algorithms that can solve large practical instances in reasonable time, whereas problems in class *P* often have.

VI. CONCLUDING REMARKS

This appendix describes some fundamental optimization algorithms that are applicable to the solution of facility location and land use problems. Of particular interest is the notion of decomposition, which can greatly accelerate an algorithm, including the case of mixed integer programs. Decomposition is broadly defined to mean the exploitation of special structure of the problem. Thus in a network-with-side-constraint model, we take advantage of the network part of the tableau by using efficient labeling algorithms in lieu of regular simplex. In this fashion, basis inverses—the more computationally intensive part of the process—are cut down to a minimum.

In Langrangian relaxation, we set aside the complicated part of the constraints and concentrate on the nice ones first. Likewise, in Benders' decomposition, we defer facing the complicated variables in preference for the better behaved ones. We bridged the gap between Lagrangian relaxation and Benders' decomposition by pointing out that both can be best illustrated by a plot of both the primal and dual space. The iterative procedure can be portrayed as a series of adjustments on the pricing scheme, represented as different slopes on the objective functions. Alternatively, the computations can be viewed as a sequence of cuts in the dual space, where each cut brings the solution closer to the optimal resource allocation. These are equally convenient and insightful ways of looking at the problem. An example of the pricing scheme can be found in Figure A4.14. There the slope of the objective functions is obtained for each extreme point defined either by $\mathbf{x}$ in Lagrangian relaxation or $\mathbf{y}$ in Benders' decomposition. An example of the cutting scheme can be found in Figure A4.15, where the dual variable is λ for the respective examples for Lagrangian relaxation and Benders' decomposition. Of equal importance here is the relationship between bounding techniques, decomposition, and duality. They represent promising analytical-solution techniques in dealing with complicating constraints or decision variables in a mathematical program.

We conclude with a discussion on computational complexity—a taxonomy to categorize algorithmic efficiency. Drawing upon the examples worked out in this appendix, we have defined the commonly used terms such as polynomial, non-deterministic polynomial (*NP*), *NP*-complete, and *NP*-hard. These terms are used extensively in the literature. Through this taxonomy, we are able to comfortably justify the efficient algorithms introduced in this appendix, showing how they compare with the more traditional approaches. While computational complexity is a useful concept, we must point out that there are issues that go beyond the categorizations. It is possible to classify algorithms and problems by their data structure. This final point is particularly relevant as we design geographic information systems to support facility location and land use models. (See Chapter 6 in this book and the "Spatial-Temporal Information" chapter in Chan [2005]).

ENDNOTES

[1] The dual of an LP is defined, according to Chapter 4, as Min$\{(\lambda_1 + \lambda_2)$: $3\lambda_1 + 5\lambda_2 \geq 1$, $6\lambda_1 + 4\lambda_2 \geq 1\}$, where λ_1 and λ_2 and non-negative dual variables.

[2] For a review of basic network-flow terminology, the reader is referred to Chapter 4, Section IV.

[3] Complementary slackness is explained in Chapter 4, Section III-C.

[4] When the entering variable is **y** instead of **x**, similar algebra applies. For details, see Kennington and Helgason (1980).

[5] LP relaxation is a common way to solve IP problems by ignoring integrality and solve the resulting LP. The integrality requirements are subsequently re-introduced through a branch and bound on the fractional variables. For a discussion and illustration of LP relaxation, see Chapter 4 (Section II-B) and the "Facility Location" chapter in Chan (2005) respectively.

[6] Notice the fourth extreme point (1, 0, 0, 1) is infeasible since it violates the complicated constraint $-8x_1 - 2x_2 - x_3 - 4x_4 \geq 10$.

[7] The terms infeasibility, dominance, and incumbency are defined in Chapter 4, Section II-B.

[8] For a complete discussion of the traveling salesman problem, see the chapter on "Measuring Spatial Separation" in Chan (2005).

[9] For a discussion of 0–1 integer/mixed integer programming algorithms, including backtracking and branch and bound, see Chapter 4.

REFERENCES

Ahuja, R. K.; Magnanti, T. L.; Orlin, J. B. (1993). *Network flows: Theory, algorithms, and applications.* Englewood Cliffs, New Jersey: Prentice-Hall.

Bazaraa, M. S.; Jarvis, J. J.; Serali, H. D. (1990). *Linear programming and network flows,* 2nd ed. New York: Wiley.

Chan, Y. (2005). *Location, transport, and land-use: Modelling spatial-temporal information*. Berlin and New York: Springer.

Fisher, M. S. (1985). "An applications oriented guide to Lagrangian relaxation." *Interfaces* (Operations Research Society of America) 15, No. 2:10–21.

Hillier, F. S.; Lieberman, G. J. (1990). *Introduction to mathematical programming*. New York: McGraw-Hill.

Kennington, J. L.; Helgason, R. V. (1980). *Algorithms for network programming.* New York: Wiley.

Lubbecke, M. E.; Desrosiers, J. (2005). "Selected topics in column generation." *Operations Research*, 53:1007–1023.

Nemhauser, G. L.; Wolsey, L. A. (1988). *Integer and combinatorial optimization.* New York: Wiley.

Reeves, C. R., ed. (1993). *Modern heuristic techniques for combinatorial problems*. New York: Halsted Press.

Winston, W. L. (1994). *Introduction to mathematical programming: Applications and algorithms,* 2nd ed. Belmont. California: Duxbury Press.

Appendix 5

Discussion of Technical Concepts

This appendix consists of technical words or concepts that are not necessarily familiar to the general audience, mainly words that are not found in a standard English dictionary. The main thrust of this book is to show the underlying concepts and the relationship between technical concepts from different disciplines. This emphasis is particularly apparent in this appendix. It complements, rather than competes with, the comprehensive index compiled at the back of this book. In many ways, the appendix extends the main body of this text inviting the curious reader to go deeper into this field. While each term is defined in as plain a language as possible, on rare occasions one technical term is explained in terms of another. When a related technical term in this glossary is used within an explanation, it is italicized, alerting the user that the related term is defined elsewhere in the glossary. Naturally, this glossary is best used in conjunction with the book index as suggested earlier.

Accessibility, impedance, propensity functions, and trip frequency curves: In this book, we discuss the importance of spatial costs in organizing the economic activities in a study area. Spatial costs are defined in many different terms. For example, spatial separation is measured in both time and cost. When we wish to convert these diverse measures into a single unit such as utiles, we face some challenges. Aside from the conventional apples-versus-oranges conversion problem, utiles is usually construed as "the more the merrier," while time and cost, or impedance in general, is defined as: "small is beautiful." To resolve this problem, we often take an inverse function of impedance to convert it from disutility to utility. This conversion function is sometimes called the propensity function, which takes on the form of a negative power function—$(impedance)^{-b}$—or an exponential function—$\exp[-\beta(impedance)]$—among others. Here both b and β are positive calibration coefficients. Irrespective of the form of the propensity function, it is usually calibrated by trip distribution curves, defined as the frequency with which a trip of certain duration is being executed in the study area. Propensity functions and trip distribution curves have similar shapes, differing only in their scaling constants. Sometimes, it is useful to normalize these propensity functions against a regional total. In a region consisting of two zones, for example, the propensity to zone 1, $\exp[\,-\beta_1(impedance)_1]$, is normalized against the total propensities to zones 1 and 2, or $\{\exp[\,-\beta_1\ (impedance)_1] + \exp[\,-\beta_2\ (impedance)_2]\}$, resulting in the accessibility to zone-1 : $\exp[\,-\beta_1\ (impedance)_1]/\{\exp[\,-\beta_1\ (impedance)_1] + \exp[\,-\beta_2\ (impedance)_2]\}$. While these terms are strictly defined here, they are often used loosely and interchangeably.

Additive versus multiplicative utility/value function: In defining a multi-attribute utility/value function, many mathematical forms can be used. For ease of calibration, it is proposed that we broadly classify the functions into two types: additive and multiplicative. The former is linear while the latter is nonlinear, with the former being more straightforward to work with than the latter. The additive form is often satisfactory when all we need is an ordinal ranking among alternatives, in other words, to rank them in decreasing order of preference. However, when preference intensity is required, or when we wish to know exactly by how much alternative A is preferred to alternative B, it is often necessary to deal with the latter function.

Aggregate versus disaggregate: A model can be either simple or elaborate, depending on the context of the study. A simple model has the advantage of transparency for policy decisions, yet often lacks the details to support actual implementation. Simplicity is often attained by aggregation where an aggregate model assumes homogeneity among the data within the analysis unit. For example, all travelers within a zone are expected to value travel time equally. A disaggregate model retains the specific valuation of each individual traveler. Aside from application, the specific analysis approach is dictated by data availability. When only average conditions are reported, an aggregate analysis is often the only feasible option. When more detailed information is available, a disaggregate model can be constructed and is often more useful. The secret is the judicious and consistent match between application, data, and models. In disaggregate land-use models, for example, individual parameters such as the labor-force participation rate (the ratio between service employment and population) can be calibrated individually for each zone, instead of for the entire region. This results in a more descriptive version of zonal level development.

Allocation or distribution: The ultimate goal of facility location or land use is to serve the demands or the customers. Thus a fire station is located for the sole purpose of putting out fires quickly, while a city master plan will provide all the services to the local population in the best way possible. The way these demands are served reflects the merit or drawbacks of a spatial decision. This service pattern is often referred to as the demand allocation or activity distribution. For example, a compact city form cuts down on commuting time from home to work. A single facility or a combination of facilities may provide the service. Sometimes, specialized services may only be provided by a facility capable of delivering such services. There may be interaction among facilities that provide these services. For example, the facilities may reinforce each other in stimulating additional demands in aggregate. At the same time, they may compete for a market share of the customers. The way the services are delivered has a direct bearing upon how well customers are served. For example, a driver and vehicle may have a specific delivery route each day and make deliveries in the order along that route, or a dedicated driver and vehicle may make an out-and-back delivery. These two delivery patterns have very different efficiencies and customer satisfaction, with the former being more efficient and the latter being more pleasing to the customer.

Area, subarea, tessellation, and Voronoi diagram: In regional science, information is often sought for each location within a study area, rather than for the entire region in aggregate. There are quite a few ways to divide up the area depending on the problem context. We may divide it up by school districts, census tracts, traffic zones or political jurisdictions, just to name a few ways. To cut across all these subdivisions, we prefer to use the word subarea, to be distinguished from the entire study area. Thus we use the word subareal population and employment rather than, for instance, zonal population and employment, since the mathematical models apply equally well across the different ways to subdivide a study area. A natural way to divide an area into subareas is to use tessellations such as the Voronoi diagram. In this representation, each activity center is defined as the *generator,* for which a zone of influence is defined, representing the demands that are naturally attracted to this activity center. It has been shown that such a tessellation is consistent with the central place theory, which hypothesizes that interregional trade leads toward natural market place settlements.

Asymtotics: A powerful tool in analysis is the asymptotic behavior of a process or procedure. Many examples of this are documented in this book. We have seen how location-routing models can be solved by heuristics that converge toward the optimum. In system dynamics, convergence and divergence of a process are analyzed in terms of *attractors* and *repellers*. In statistics, the central limit theorem suggests that many probability density functions converge toward a normal distribution given a large enough sample is taken. In spatial processes, however, things are more complex. The central limit theorem may not apply. This prevents borrowing ideas from classical time-series models and applying them wholesale toward spatial processes. In our discussions in this book, we have tried to highlight both the commonality and differences between spatial statistics and classical statistics. We draw analogies where we can between them, and at the same time point out some unique behavior of spatial information not found in classical statistics.

Attractivity and lattice structure: The two paradigms used in facility location and land use are *discrete* versus *continuous* modeling. Irrespective of the type of models used, there exists a central location where activities are generated from or attracted to, which takes on such names as *centroids* or *generators*. The system of centroids or generators often form a lattice structure, which shows the proximity and relationship between them. Now the demands within the zone of influence of a centroid or generator are expected to interact with the respective centroid or generator. The zones of influence or market area can be defined by a Voronoi polygon or other representation. The interaction between centroids or generators can in turn be described by a number of different models. Among them are the gravity model, the weight matrices or *spatial masks,* or the Delaunay triangles. The gravity model suggests that nearby and more intense activities interact more than distant or low intensity activities. The spatial mask simply summarizes these inter-centroid interactions in a matrix form. Should spatial interaction be modeled by tile-like structures such as Voronoi diagram, the Delaunay triangles are simply the most direct paths of interaction between generators.

Attractor, repeller, and separatrix: A dynamic system is said to be stationary when its behavior does not change over time. Close study of the stationary behavior reveals some commonalities among them. The trajectories can be attracted to a sink or a focus, reaching a stable solution. Alternatively, the trajectory can be repelled from a focus or they can be separated at a saddle point, suggesting instability. A sink and stable focus are categorized as an attractor while an unstable focus and saddle point are called a *repeller* and a *separatrix* respectively.

Average versus marginal cost: Average cost is the total production cost apportioned among the number of units produced, and marginal cost is the cost of producing an additional unit. A firm is most efficient, and is most profitable, if it produces at the level where marginal revenue (the unit price charged to consumers) is equal to marginal cost. In contrast, it would be less efficient and only breaks even when marginal revenue (price) is set equal to average cost. While these basic definitions are well understood, its application in the public sector is more controversial. An example is marginal cost pricing of transportation services. It can be shown that it prevents Braess' paradox from happening, or that individual trip cost is worsened after a new link is added to the network. In this regard, marginal cost pricing is desirable in that the average travel cost for the user is less. Marginal cost pricing, however, may result in less total usage of the transportation facility. Another example showing the controversial effects of pricing is in the development of a new community. Accessibility may be provided to a new community at the expense of the existing communities. But consumers' surplus is increased for the entire population. In all these examples, the basic average and marginal cost concepts carry over to transportation when spatial equilibrium is considered. Notice that transportation cost is added on top of production cost in determining interregional trade. In addition, demand for transportation is modeled as a function of the transportation cost. Trade occurs when transportation cost is compensated by the difference in price between the production point and the consumption destination.

Backshift operator, transfer function, lag operator, and image processing mask: In analyzing time series data, the backshift operator is a useful concept. It affords a compact notation and tool for writing and analyzing time shifted data. In its simplest form, the backshift operator simply lags a time series by one or more period. Linear combinations of these lagged series are then formed by putting different weights among each series. In a more involved usage, algebraic manipulations such as divisions can be performed on these operators, allowing powerful analysis of time series. In a similar vein, a transfer function consists of a set of weights placed upon an input time series and transforms it to a different output series. Often, the input series is white noise, or uncorrelated random shocks, and the output series consists of correlated information. If the input series is not white noise, its underlying pattern can be taken away by prewhitening the time series. When these concepts are applied toward spatial data, we have the spatial lag operator, which performs a similar function to that of the backshift operator, except in two dimensions. In image processing, we call this a mask, defining how a subject data point (such as a pixel) relates to its surrounding data (or "next door neighbors"). An image processing mask consists of a set of two-dimensional spatial weights to be applied toward each data point. This transforms the two-dimensional data set (an image) into a different one, often making it less noisy or giving it more definition.

Basic versus nonbasic activities: According to economic-base theory, basic activities are the goods and services that are seeds of economic development for a local area. They are generally consumed outside the study area. Nonbasic activities are derived from basic activities. They are the result of the multiplier effect of basic activities upon the local economy. For that reason, nonbasic activities are for local consumption. The difficulty with this paradigm, however, lies in how defensible these definitions of basic versus nonbasic activities really are. While the fundamental concept upon which the Lowry model is built can be traced to economic-base theory, the explanation becomes blurred once subareal allocation of activities becomes a key element. The distinction is further complicated by economic development over time. In the local versus outside world paradigm of economic-base theory, it is fairly easy to distinguish between export versus local consumption (the location quotient definition), or the requirements one economic sector places upon another (the minimum requirement definition). However, it is not so clear in applying this concept to subareas in the local economy over time, where basic employment is supposed to be exogenously fixed, while other service employments are located within the study area in relation to these basic activities. The interaction among these activities can relocate basic employment. Development over time can also change the requirement one economic activity places on another, particularly when the zonal level of detail is required.

Bayesian or subjective probability: According to Bayes, all probabilities are subjective. However, the more sampled information one has, the better one can determine the underlying probability with precision. Classic decision analysis builds upon this fundamental idea and extends it into multiple attributes in recent years. In remote sensing, this concept is used in classifying pixels in a satellite image of land cover, to discern whether a pixel belongs to a lake or a forest for example. In time series, it is used to update the mean, variance, or model form as new data become available. In accordance with Bayes, while we can get a good approximation, there really is no practical way to obtain the true probability distribution.

Capacitated versus uncapacitated: In facility location and land use models, distinction is made between whether or not a holding capacity exists in a geographic unit. The model tends to be a great deal simpler to solve if there is no capacity, but this is often a simplifying assumption. The demand increments are then allocated or distributed exclusively to the closest facility. However, when capacity is present, demands or activities above and beyond the facility or zonal capacity must be assigned somewhere else. No longer is there a binary pairing between demand and facility (or zone). Fractional assignments take place, in which only part of the demand is accommodated by a single facility. In other words, more than one facility is assigned to a demand.

Cardinality versus ordinality: In ranking alternatives, we can be satisfied with a simple preferential ordering, in which the ones from the top of the list are picked over those at the bottom. If the preferential intensity is required, or we wish to assess by how much alternative A at the top of the list is better than B at the bottom, a scale or cardinality is involved. Obviously, the former preference structure is simpler than the latter since less information is required to capture the preferential intensity.

Catastrophe, hysteresis, and divergence: The trajectories of a chaotic system can be categorized. Among them is the sudden jump (or catastrophe), hysteresis, and divergence. A sudden jump is fairly self-explanatory. Hysteresis is defined as a trajectory in which a path, if reversed, ends up at a point other than the starting point. A divergence suggests that a small difference in approach leads the system to a very different state. These counter intuitive phenomena are different manifestations of a catastrophe theory in general.

Center versus anticenter: A center is the facility location that ensures that the furthest demand is kept closest. The best example is a fire station, which should be close to any fire that may flare up even at the most remote location. An anticenter, on the other hand, is a location that keeps the closest demand as far away as possible. Thus it is desirable to put a landfill as far away as possible from the most exposed residence. Formally, we minimize the maximum distance between demand and the facility in the center problem, and we maximize the minimum distance in the anticenter problem.

Centroid or generator: Two paradigms are used throughout facility location and land use: a geographic representation can be continuous or discrete. One way to compromise discrete versus planar models of land use is through the use of centroids. Population and employment distribute continuously over a map. However, there is advantage in modeling the zonal or subareal population or employment as concentrating in a single node called centroid. Centroid is an imaginary node/vertex amid a plane through which activities (for example, trips) originating from or destined for the subarea are loaded or unloaded from the map. In this case, the centroid is the geographic center or center of gravity for the economic activities in this zone (or subarea). The use of centroids can also be thought of as a more aggregate representation than its counterpart. Viewed in this light, the question really boils down to how big one should define a zone (or subarea), and correspondingly the accuracy of using relatively few number of centroids (vis-a-vis the extra computation involved with a larger number.) In this text, centroids and generators are used interchangeably. The word generator comes from the concept of Voronoi diagrams, representing the natural gathering place for a subarea. (See also *internodal* versus *intranodal*.)

Collinearity: When two random variables are related to one another statistically, we say that they are collinear. Problems arise when both of these two variables are included as independent or explanatory variables in a regression model. This amounts to double counting the same explanatory variable in a model. The result may be spurious correlation, or statistical relationships that are chancy and not well supported by the facts and figures.

Combinatorial: Among discrete alternatives, one way to account for the various options available is by combining them in different ways. In obnoxious facilities location, for example, a combination of landfill, incinerator, and transfer station may be desirable to dispose solid wastes. When there are many alternatives, the combinations can become huge very quickly. Finding an efficient way to choose among these combinations is mandatory. Modern mathematics has provided many guidelines to achieve this goal, as evidenced by recent advances in combinatorics, the branch of mathematics that addresses this type of combinational problem.

Complementarity problem: An optimization problem is often characterized by complementarity conditions, or the relationship between the primal and dual representation of the problem. For both linear and nonlinear optimization, the gradient of an objective function or functional is orthogonal to the feasible convex set at the point of interest. In other words, the product of the gradient and the feasible solution is always zero. For example, in the first row of a simplex tableau, the rate of ascent (gradient) for a particular basic (non-zero) variable is zero, while the rate of ascent for a nonbasic (zero) variable is non-zero. In both cases, the product of the two pertinent quantities—gradient and variable—is always zero. In facility location and land use, spatial price equilibrium, or the way that trade takes place between any two points, can be best formulated as a complementarity problem.

Complementarity versus substitutionality: According to classic microeconomic theory, goods or services are either complements or substitutes of one another. In our context, users either view alternative facilities as catering to their needs in an aggregate or the goods and services offered by individual facilities are replaceable among themselves. In most cases, the facilities play both roles, although not equally. The relative dominance among these roles results in interesting spatial-activity derivation and allocation results that serve as extensions and generalizations of classic facility-location models. These extensions and generalizations eventually bring us a land use model.

Condorcet versus Simpson points: In discrete facility location models, a Condorcet point is any point in the network that is closest to most of the demands. A Simpson point is the least objectionable place where the maximum demand closer to another point is at its minimum. Both Condorcet and Simpson points are relative, rather than absolute, concepts.

Contextuality: In discerning spatial patterns, whether they be land use or satellite images, one can obtain a fair amount of information by observing the neighborhood of what one is examining. Thus a noise pixel can be detected quite clearly as an outlier and subsequently removed when context is taken into account. Similarly, a facility location decision cannot be divorced from the community in which the facility is to be sited. In this latter case, the local context of the problem drives the location decision.

Control, state, slow, and fast variables: The term decision variables used in operations research becomes control variables in control theory. Dependent variables, on the other hand, are labeled state variables. In using catastrophe theory to analyze system stability, control and state variables are called slow and fast variables respectively. When more than one control variable is present, we refer to the set as control point.

Convex versus nonconvex programming: In optimization, one has either a single (unique) global optimum or more than one local optimum. The former is associated with a convex mathematical program while the latter a nonconvex program. A convex program is characterized by a number of nice mathematical properties that satisfy strict *duality* conditions. On the other hand, nonconvex programs are characterized by less rigorous conditions, making the solution algorithms more ad hoc. In the majority of network facility-location models, integrality (or discreteness) dictates an integer programming formulation, which is nonconvex. For this reason, facility location models, besides being challenging to apply, are also at the frontier of discrete optimization research.

Cross-sectional versus time series data: In the context of spatial-temporal information, data points within the same time period are called cross-sectional data. A time series, on the other hand, is the tracking of spatial data over multiple periods. In household terms, the former can be thought of as a snapshot, while the latter is a movie.

Curvilinear: An efficient frontier is defined as the set of nondominated solutions when judged in terms of two or more criteria. In competitive or group decision making, an efficient frontier can either be linear or bounded by curved lines. The latter, or the curvilinear case, poses more computational challenge than the former.

Dependent versus independent, criterion versus predictor, endogenous versus exogenous variables: In regular regression, a dependent variable is explained statistically by a number of independent variables. The dependent variable appears on the left-hand side and the independent variables on the right-hand side of the equation. Independent variables are also called explanatory variables or regressors. The dependent variable is sometimes called response variable. In canonical correlation, one ascertains if a set of criterion variables are possibly affected by a set of predictor (input) variables. Often, the number of criterion and predictor variables can be collapsed or reduced to enable a simpler, more tractable analysis. In econometrics, simultaneous equations are used to pose the relationship between a number of endogenous variables and exogenous variables. Endogenous variables appear on both the left-hand side and right-hand side of the equations, while exogenous variables only appear on the right-hand side.

Deterministic versus probabilistic: Traditionally, the location literature has been modeling spatial analysis in a sequence of deterministic events. Thus we know exactly where the demands are, and we locate a service facility that will be proximal to these demands. Recent decades have witnessed a broadening of view when the demands are no longer known precisely ahead of time. Rather, they are random or probabilistic. For example, a fair amount of progress has been made in recent years in locating fire stations when there is little knowledge about when and where a fire may break out. In this example, it is not easy to lay down a set of rigid rules one follows in selecting a fire station site. One way to site a fire station is to think of all possible ways fires can break out and locate the fire station such that it will respond to these fires the fastest way. This is obviously not a straightforward process and may take a huge amount of computer time (if it is at all possible.) A better way is to model the fire outbreaks in terms of a random process (stochastic process) and marshall our knowledge on random processes in modeling the situation. The challenge obviously is to carry this in a spatial context, posing additional analytical intricacies beyond the difficulty with modeling standard random processes.

Differencing: Finding the change in values among discrete points in a grid is the numerical equivalent of finding the differential in continuous variables. It is a much more general technique in that we can solve a much larger class of problems, particularly when assisted by modern day computing. In spatial analysis, it is often used to restore and enhance an image, such as in edge detection. Statistically, it also induces a *homogeneous* set of data, which allows for more insightful analysis to be performed (see separate entry in this glossary for the definition of *homogeneity*). In time series, data are differenced to achieve *stationarity,* for similar purposes. Differencing in this case removes the trend from the data, resulting in a constant mean. It allows us to concentrate on discerning the underlying pattern in the data.

Dimensionless analysis: Oftentimes, it is insightful to display analysis results that are independent of the physical units used. An example is the nomographs used in queuing handbooks. For comparison among queuing disciplines, it is most insightful to emphasize the relative performances among these queuing disciplines without regard to whether a metric or English system of weights and measures is used. Instead of worrying about measuring total time in the system in minutes, hours, or seconds, it is best to display it as a multiple of the service time. A total-time-in-the-system being one means the only time required is the service time and the system is totally congestion free. No wait is involved in this case, and the customer receives service right away. On the other hand, a total time bigger than one would suggest congestion, in which some wait in line is inevitable. The amount of wait is again quantified in multiples of the service time. Queuing delay of 0.5, for example, means the customer waits in line half as much time as being served.

Discrete versus continuous: Distinction is made between continuous and discrete variables. When a variable assumes integer values such as 1, 2, 3 . . . , for example, we say that it is a discrete variable. On the other hand, when the variable can assume any rational value to as accurate a decimal point as needed, we call this a continuous variable. In facility location and land use, we can either locate the facility in a network of arcs and nodes/vertices or on a plane. When a facility is to be located in a network, it ends up at a node/vertex or on an arc. We call this discrete facility location. In contrast, if a facility can be anywhere on a plane, we refer to it as a planar location or continuous facility-location problem. Ranking discrete alternatives such as the candidate sites for an airport is a complex task. Unlike its continuous counterpart, multicriteria simplex procedure is no longer valid. We may miss some discrete alternatives that form the efficient frontier. Under these circumstances, implicit enumeration among pairs of alternatives is a viable solution option. ELECTRE, for example, is a computer program advanced by Roy (1977) to rank-order a set of discrete alternatives. This was referenced in the bibliography of Chapter 5. Consisting of a graph theoretic outranking procedure, it identifies the set of noninferior solutions. Since its inception the software has gone through at least two new releases to the public.

Discriminant: A yardstick is often required in classifying a population into groups. This decision boundary is often represented in a discriminant function, which includes the important attributes that distinguish one group into another. For example, a properly defined discriminant will allow us to tell whether a picture element (pixel) belongs to a lake or a forest.

Disjunctive graph: In a mathematical program, a set of constraints is called disjunctive if at least one of the constraints has to be satisfied but not necessarily all. Consider a multiple traveling salesmen example, there are n demand locations to be visited. To cover each location i, m salespersons are to be used in a given order. The total time of the visit at location i by salesperson k is finite and is known. The problem consists of finding a fixed order of visiting the demand locations sequentially by each salesperson so as to finish visiting these n locations as soon as possible. For a given salesperson k, the tours (i, k) for $i = 1, \ldots, n$ can be represented in a potential task graph called a disjunctive graph (a clover leaf graph in this case). Here there are m such graphs (clover leafs), one by each salesperson.

Dissipative structure and self-organizing systems: The term dissipative structure stems from physical systems with a permanent input of energy that dissipates through the system. If energy input is interrupted, the system collapses to its equilibrium state. This stands in contrast to conservative dynamical systems in classical mechanics. In a conservative system, there is neither an additional input nor a loss of energy, implying that no friction exists. As part of the development of socio-spatial dynamic theory, G. Nicolis and I. Prigogine (1977) proposed a theory of self-organization that was observed in phase transitions in physical chemistry [*Self-Organization In Non-Equilibrium Systems* (Wiley, 1977)]. Departing from conservative systems, they illustrated various self-organizing and non-equilibrium systems well beyond physical sciences, ranging from dissipative structures to order through fluctuations. (See also *equilibrium* and *disequilibrium*.)

Duality: In facility location and land use, duality has several meanings. The first, perhaps the simplest, is the mathematical programming usage of the word. It provides everything from computational bounds to economic interpretations. An example is the game theoretic interpretation, as in simple games that can be analyzed as a primal-dual linear program. Primal and dual variables give significant insight into location problems. Dualization of a mathematical program also allows more efficient algorithms to be implemented. Then there is the application of duality in spatial tessellation, or the analysis of space in terms of tile-like units, ranging from squares to polygons. A dual graph can be constructed for every tessellation that is presented, giving significant insights and allowing for computational savings in location decisions.

Duopoly, triopoly, quadropoly, and oligopoly: In competitive facility location, each provider is locating a facility to capture as large a geographic market share as can possibly be managed. When there are two competitors, we have a duopoly. When we have three competitors, we have a triopoly. When there are quite a few number of competitors, we have an oligopoly. Unlike monopoly or pure competition, oligopolies are quite complex. While an oligopolistic market is challenging to model to begin with, the spatial version of it certainly does not make it any easier. It turns out that the land-use modeling literature has a much richer knowledge base to offer than the discrete/network facility-location analysts on this subject. It represents an area where the land-use and facility-location models may draw synergistic benefits from one another.

Econometric(s): The discipline of economics used to be quite a bit more qualitative than it is today. Recent years have witnessed tremendous emphasis on quantifying a number of concepts commonly used in classic economics, including estimating demand and supply functions. As an adjective, econometric describes any undertaking in estimation and measurement. As a noun, econometrics is the science and art of estimation and measurement. This typically involves analyzing historical information in support of a statistical hypothesis. There are two common types of land use models, one is based on deterministic simulation and the other on econometric models. There is a relationship between time series and econometric models wherein a specialization of the coefficients in a multivariate time series yields an econometric system of equations.

Eigenvalue/eigenvector or characteristic-value/characteristic-vector: A model in equilibrium is often described by a system of homogeneous linear equations. The behavior of the model is characterized by a parameter, which we call the eigenvalue. In a mechanical system, for example, the eigenvalue is its natural vibration frequency. In a multicriteria decision-making model such as the analytic hierarchy process, the principal eigenvalue measures the consistency with which the pairwise comparison survey is completed by the decision maker. The eigenvector here is the set of weights the decisionmaker places upon each attribute as implied by the completed survey. In adjusting a time series to change, the eigenvalues of the estimation-error variance-covariance matrix may also be required. In this case, the correlative properties of the time series are captured in the variance covariance matrix.

Elliptic, hyperbolic, and parabolic umbilic: As the control variables change, a system can transition from a stable to unstable pattern at bifurcation points. One type of such transition can be described by an elementary catastrophe called an umbilic, which geometrically suggests a depression in the center of a surface through which potential can be transferred. Depending on the number of control and state variables, we can have an elliptic, hyperbolic, or parabolic umbilic. The former two are variations of the parabolic umbilic, obtainable, say, by replacing the potential function by its negative. As with other elementary catastrophes such as the cusp and the butterfly, they are canonical models rather than actual description of the system under study. They allow us to understand the qualitative behavior, rather than the quantitative behavior, of catastrophes in the system being studied.

Emittance: Most remote sensing devices work on signals that are reflected off the object being observed. We say that they process the emittance from these objects. The emittance is different depending on the reflective angle, and the type of energy source used to illuminate the object. The emittance data are often processed and filtered for best detection by selected sensors.

Entropy: Borrowing from its Greek origin meaning "change," entropy is best interpreted in the spatial context as a measure of the frequency with which an event occurs within a closed system. In an aggregate statement of a travel pattern, for example, it is sometimes useful to have a description that is robust enough to accommodate as many possible detailed patterns as possible. This is called entropy maximization, and is applied often to capture all possible patterns. In the absence of any additional information, this results in equally likely travel toward each destination given a fixed number of trips emanating from a central location. Any additional information would obviously modify the homogeneous travel to a pattern other than uniform, to be consistent with the newly acquired knowledge. Additional information would include such knowledge as the relative trip lengths, which provide the percentages of short, medium, and long trips. When interpreted this way, entropy maximization is equivalent to the *information minimization* principle. (See also *micro, meso* and *macro* states.) Entropy can also be interpreted as spatial uncertainty. It measures the degree of diversity in the dominance of destinations. For a regular triangular lattice of equal size and with no boundary effect, spatial uncertainty is at a minimum at the grid points and at a maximum in the intervening space.

Enumeration: One of the ways to identify the best alternative is to examine each and every alternative. Comparison among their figures of merit will reveal the best alternative. For example, we may wish to select the least costly alternative. In the real world, however, such enumeration is either impractical or impossible due to the large number of alternatives that exist. Here is when a mathematical model of the problem may become useful. Solution to the model automatically sorts out only the most promising alternatives or the very best alternative in an efficient way, without having to enumerate them exhaustively.

Equilibrium versus disequilibrium: Equilibrium is a stable state of a system in which there is no immediate tendency for change. Small perturbation would not dislocate the equilibrium state. The opposite situation is disequilibrium, in which the system is characterized by instability. Disequilibrium can take on many forms, including a cyclic pattern and truly chaotic patterns that do not have any discernible order. (See also *dissipative structure* and *self-organizing systems*.)

Exponentiation: An exponent is the power to which a mathematical variable or a mathematical term is raised. To exponentiate is to raise the entity to its power. In spatial-temporal analysis, spatial cost is often measured in terms of an exponentiated function of distance or time. For example, an exponent of one gives a linear spatial-cost function; an exponent bigger than one a convex function, and an exponent of less than one a concave function. Whether a unique optimal location is obtained or where it is found depends on the shape of this function. It also turns out the value of this exponent can transform one class of spatial problem to another seemingly unrelated class. (See also *parameterization*.)

Externality: Microeconomics accounts for the transactions between various parts of the economy via the price system. The price system is the mechanism by which supply and demand of goods and services are cleared in the marketplace. It becomes quite evident that not every transaction can be regulated by price. An example is pollution, which industries often incurred as part of the production process. Yet its cost to society in terms of health hazards is not often charged toward the industry. These costs are external to the accounting system and therefore unaccounted for. We call these externalities.

Extremal and extremal solution: A well-known fact in linear programming is that the optimal solution has to occur at an extreme point (or corner point) of the feasible region. This property is carried over to network facility-location models. For example, the optimal location is often found at a node/vertex (or an intersection) of the street network at which a fire station is to be sited. This is not an intuitive result by any means, since there is no reason a priori why the optimal location cannot be on an arc or at any other place. This nodal optimality property, where identified, does allow us to design some computationally efficient solution algorithms. Available evidence suggests that certain extreme conditions also exist in planar location models, in which the facility can be sited at any point in the Euclidean space. For example, the optimal airport among three cities is often located at one of the cities. In other words, the optimal site is at a vertex of the triangle formed by the three cities, rather than somewhere inside the triangle. In a calculus-of-variations problem, or a special case of a control problem, the solution for an optimal path that satisfies the initial and end conditions is called an extremal or a stationary function. This usage should not be confused with extremal point optimality mentioned above.

Factorization: As explained under *frequency domain*, Fourier transform is a convenient analysis of a signal over time. In Fourier analysis of discrete, mass probability distributions, the transform is expressed as, or factorized into, a polynomial of functions of complex variables. Unfortunately, this cannot be carried over to the spatial domain directly. In a random or Poisson field, joint probability mass functions can be factorized into conditional probabilities only under stringent positivity conditions, where the positivity condition is a prerequisite property for a random field. Under this situation, conditional probability models for data of this kind cannot be of the simple nearest-neighbor variety commonly used to analyze spatial data.

Feng shui: In Chinese mythology, facility location should fit into the harmony of the natural environment. This includes orientation of the facility with respect to the topology and layout of the surrounding land. Literally translated, feng means "wind" and shui means "water," referring to the elements. Should a facility be placed the wrong way, bad luck will follow; while proper placement will bring good luck. Increasingly, the western world has caught on to these qualitative factors in facility location. The idea of feng shui is introduced in this book to highlight its scope (and limitations). Instead of using a holistic view like feng shui, we are often concentrating on the effect of one factor at a time. For example, what is the effect of highway construction (specifically) upon facility location and land use?

Fractiles and fractile method: Fractal comes from the Latin word fractus, meaning "broken," describing objects that are too irregular to fit into traditional geometrical setting. Many fractiles have some degree of self-similarity, they are made up of parts that resemble the whole in some way. The similarity may be approximate or statistical. A space-filling curve used in routing is an example of fractiles. The space-filling curve transforms a two-dimensional map into a single dimension. By observing the clusters in the single-dimension line instead of proximity in two dimensions, vehicle tours can be constructed much more conveniently for each cluster of demand points. Fractile method is really a very different concept altogether. The word fractal is used only because we split a line up into fractions in this method. In constructing the univariate utility function, one common way is to have the decision maker play a lottery. The objective is to locate an indifference point by which the decision maker is undecided between playing the lottery and being awarded a fixed sum. Based on preference for the lottery or the fixed sum, the decision maker is either identified as risk-prone, risk-neutral, or risk-adverse, and the corresponding convex, linear, or concave function defined. When enough lotteries are played, a sufficient number of points are obtained to plot the univariate function. Drawing upon the common coin tossing experience, it is natural to design a 50-50 lottery. Such a lottery also has the nice property of dividing the vertical axis of the univariate utility function into halves, quarters and so forth each time an additional point is defined on the curve. Such a survey procedure is referred to as the fractile method.

Frequency domain, Fourier transform, line spectrum, periodogram, and time/image domain: A signal or data emitting from a source can be analyzed in a couple of ways. We can analyze the signal directly (in its time domain), whether it be a time series or a spatial image, or we can examine its frequency. Each of these two methods has its advantage. The time domain is more intuitive, since it describes the signal directly. The frequency domain, typically represented in terms of a line spectrum or periodogram, is convenient for noise removal, as noise has a distinctly different frequency than a regular signal. By examining the line spectrum or periodogram, outlying frequencies corresponding to those from noise, can be easily discerned and removed. Fourier transform is a common technique used to analyze the frequency of the signal and reconstruct the signal once the noise is removed in the frequency domain. Among other uses, seasonal data patterns can also be easily picked out from a line spectrum or periodogram. This will help in identifying the correct time series model. Fourier transform is a convenient way to analyze signals in the frequency domain. In two-dimensional images, there are parallel, and sometimes more superior techniques for performing similar functions when stringent assumptions are made. Not only is the noise removed in this case, often the image is made more crisp also by virtue of sharpening the outlines.

Gaming: Many factors in facility location and land use cannot be quantified precisely, particularly the rivalry between stakeholders. Although the state of the art has progressed significantly, the analytical techniques advanced in this text—including game theory—are limited in their utility. This is the reason why we discussed gaming. Gaming is an exercise that immerses the interested parties in a replica of the real world scenario. Divorced from the dangers of failing, a player can step through the many faceted situations of spatial decisions and learn from the experience in a game. CLUG, the Community Land Use Game, is one such game. Originated by Alan Feldt (1972), the game simulates the real world of land development, complete with monetary transactions, urban renewal, and politics. (See reference in the bibliography of Chapter 3.) It is a forerunner of many subsequent efforts in this area.

Gradient versus subgradient search: Gradient search is the most general way to solve regular nonlinear optimization problems. It is also known as the method of steepest ascent/descent. When the slope is not smooth, but piecewise linear, an equivalent scheme, called subgradient search is employed. In the former case, a gradient is computed in each step. In the latter case, a Lagrangian relaxation problem needs to be solved first to determine the subgradient. In both cases, a step size is computed to show the distance along which one climbs the slope. Both algorithms terminate when either the gradient or subgradient approaches zero.

Graphs and networks: Graphs are convenient, visual ways to represent the relationships between land use entities such as population and employment. For example, employment opportunities will bring in dependent population into the community. This can be represented as an arrow drawn from employment to population. Other relationships can likewise be sketched, from which mathematical models can be constructed. An advantage of such representation is that certain mathematical properties can be readily discerned in the graph. In constructing an econometric model between population and employment, for example, we can easily postulate the correlation coefficients between variables. The graph helps to identify the expected values of some of these correlation coefficients. When flows are introduced in directed graphs, or a graph with arrows drawn on the arcs, a network is obtained. Again, a network can unveil useful mathematical properties of the model. An example is the representation of a facility location model as a network-flow model. Here, the incidence relationship in a tree graph is directly equivalent to the basis matrix of the corresponding linear programming formulation of the facility location problem. Instead of inverting a basis matrix, we can now accomplish the same thing by manipulating the graph. In this book, we show that many facility location models can be solved more efficiently when represented as a network (rather than as a formal mathematical program.) Also useful are the computational properties of the network constraints, which lends itself to an integer solution for integer right-hand sides of the mathematical program.

Homogeneity versus heterogeneity: Spatial data are said to be homogeneous if the statistical inference made at unit *i* is the same irrespective of where *i* is. On the other hand, the data are heterogeneous if this is not so, or the inference is different dependent upon where *i* is. The stochastic-process approach to spatial data means that only one observation is variable at each instance. In other words, the process of allocating values to the random variables in space or space-time is performed one at a time. This gives rise to a computationally imposing situation, and there are some operational difficulties. Since this is not particularly practicable, some restrictions need to be imposed on the degree of dependence and heterogeneity that can be allowed. Only in this way can one handle the spatial stochastic process. Essentially, in order to infer certain characteristics of the underlying process, a degree of stability—such as homogeneity—needs to be assumed among the spatial data.

Homoscedasticity, stationarity, ergodicity and isotropy: In ordinary-least-squares regression, the model is homoscedastic if the residuals are uniformly distributed about the dependent variable means as shown by the regression line. In time series, where we regress a series against its lagged series, the same property is named stationarity. Similarly, a time-varying (stochastic) process is said to be stationary if it has become regular in its behavior (or reached a steady state). In this situation, the dependent variable would have a constant average value. Often, we prefer to model the underlying stationary process (rather than, say, the evolving process or the raw data) for a couple of reasons. First, it is more insightful (and therefore more valuable) to understand the underlying behavior. Second, it is easier to model than its non-stationary counterpart. Once the underlying process is understood, we could always map the results back to the dynamics of evolving process. When spatial data are involved, it is often advantageous to model it as a random or Poisson field. While stationary concept still carries over in general, the process is much more complex. One other useful property here is ergodicity, a concept borrowed from a memoryless random process called Markov chain. Ergodicity ensures that, on the average, two events will be independent in the limit. Now recall that by definition a Markov chain is ergodic if all states in the state transition chain are recurrent, aperiodic, and communicate with each other. An ergodic assumption allows for consistent estimation of the joint probability of various variables in a spatial time series.

Hypercube model: The model dispatches a fleet of service vehicles in response to calls. A vehicle at a depot is either free or busy, as represented by the binary 0–1 variable. For two depots with a vehicle at each, (0, 0) denotes both vehicles are free and available for service, (0, 1) means only the vehicle from the first depot is free; (1, 0) means only the vehicle at the second depot is available; (1, 1) says both are busy. The four states of the system—(0, 0), (0, 1), (1, 0), and (1, 1)—can be plotted as four nodes/vertices in a graph that describes the possible transitions between these states. Such a state transition graph resembles a rectangle, characterized by the four nodes/vertices and arcs representing the possible transitions between the states. When there are three depots, the graph resembles a cube. In the general case when there are any number of depots, the graph is a hypercube, and hence the name hypercube model. Technically speaking, it is a spatial queuing model that caters to random calls or demands at an average arrival rate.

Inflexion point: Change of a graph from convex to concave or vice versa. In the context of a simple elementary catastrophe, the inflexion point could show the transition from stable solutions to unstable solutions. Thus in a plot of the *functional* against a control variable, a negative control variable may signify the existence of stationary solutions, while a non-negative value signifies instability.

Infrastructure: The functioning of society is supported by a number of facilities that are critical. Examples include utilities, transportation, and water supply. They constitute a web of basic building blocks essential to a standard of living. This book is concerned with the judicious configuration of such facilities, or infrastructure, in achieving certain goals.

Integrality: Many spatial-temporal models require the decision variables to assume binary or integer values. For example, we either locate a facility at a node/vertex or we do not, a decision often represented by a binary 0–1 variable. Similarly in image processing of satellite photos, we either classify a pixel (picture element) to belong to the lake or the forest, but not to both. Unfortunately, the computational requirement to solve this type of problem is often explosive. This requires careful model formulation as well as fast algorithms, not to say advanced computational machinery. When formulated as a binary or integer program, many nice mathematical properties associated with, for instance, a continuous variable model are also absent. For all these reasons, integrality requirements are challenging (and often impossible) to fulfil. In locating a facility on a network consisting of nodes/vertices and arcs, the optimal location is often found at a node/vertex. This is a desirable property since it saves computational efforts. Nodal optimality can be thought of as an analogue of the familiar extreme-point optimality condition for linear programming. Both nodal optimality and extreme point optimality are not obvious in many models and a fair amount of attention has been paid by researchers to identify the conditions under which nodal optimality holds. Nodal optimality conditions can be identified in median, center, deterministic and stochastic facility location problems (See also *extremal solution*.)

Internodal versus intranodal: In spatial representation, approximation is often necessary. Thus we may consider all the population or employment in a zone to concentrate at a node (often called a centroid), while in reality, they are distributed among every part of the zone. Under this abstraction, trips executed by the residents or employees will take a finite amount of time to come out of the origin node in their journey toward a destination. Once they are in the destination zone, it will also take a finite amount of time to get to its ultimate destination. We refer to this finite egress and access time as the intranodal travel time, with the time covering the line haul journey from origin to destination zones as the internodal travel time. (See also *centroid*.)

Interregional transactions: Much of economic development and land use is concerned with the trade between geographic regions of interest, including imports and exports. Interregional transactions form the driving force behind spatial evolution. It is not sufficient only to model trade between economic sectors such as manufacturing, service, and household. Much of our concern here in this book is on where these manufacturers, service providers, and households are located, since their locations determine how much interaction is expected between them.

Intersectoral transactions: An economy is made up of sectors such as the manufacturing sector, the service sector, the household sector, and so on. Each sector trades with another in the conduct of business. Thus a manufacturer purchases auditing service from the service sector and hires labor from the household sector. Similarly, the household sector purchases manufactured products from the manufacturer and buys entertainment from the service sector. This results in intersectoral transactions, which in turn makes the economy go round and round.

Intransitivity: To the average person, if alternative A is preferred to alternative B and alternative B is preferred to alternative C, then alternative A is preferred to alternative C. Contrary to intuition, however, such transitivity between alternatives does not necessarily hold. While many such transitive cases exist, the world is replete with intransitive alternatives. One can easily construct an example that under a democratic voting process (based on majority), cyclic ranking can result. In this case, A is preferred to B, B is preferred to C, and C is in turn preferred to A. We say that intransitivity is observed. Intransitivity often arises when one is judging along conflicting stimulus dimensions.

Isocost, iso-utility and isoquant curve: For a fixed amount of capital outlay, various combinations of resources can be purchased. The tradeoffs among these resources form the isocost curve. Thus for a fixed household budget, one may wish to trade off between spending it on housing and transportation. Living further out of town will presumably lower housing cost, but this is done at the expense of higher commuting expenses. The isocost curve forms the frontier of the purchasing power of a fixed budget outlay. The household settles on a combination that maximizes its aggregate utility. The combination is often determined by an indifference curve on which the household gets equal pleasure on each point on the curve. Thus the curve represents a constant utility to this household. Viewing from the producer's side, certain combination of input factors, such as labor and raw materials, will achieve a certain level of production. Several combinations of input factors will accomplish the same level of production. The line drawn linking these combinations is the isoquant curve. A market equilibrium is then determined by the consuming households and the producing industries.

Lagrange multipliers, dual, costate, or adjoint variables: In an optimization problem, it is often of interest to impute the marginal value of a resource. Various disciplines have different terminology for the same concept. Economists may call it the opportunity cost. Mathematicians call it the Lagrange multiplier or dual variable. Control theorists call it the costate or adjoint variable. The Lagrange multiplier or dual variable is usually associated with the relaxation of a limited resource, whether it be a budget or other constraints. It answers the question: What will another dollar in the budget buy me when my budget has been exhausted. Costate or adjoint variables, on the other hand, refer to the marginal value of a stock. Thus for an inventory problem, this amounts to the opportunity cost of a unit of inventory shortage.

Linearity: In its simplest form, a linear function has its dependent variable directly proportional to the independent variable. A linear system does not have reinforcing effects among the inputs, in other words, the response is directly proportional to the applied excitation. A linear operator has the property that the effect on the sum of two components is similar to that of each component. A linear filter, for example, takes a weighted sum of a time series to transform it into another time series. Thus a simple filter may just delay a time series by a constant number of periods. These linearity properties allow superposition of the effects of each of the individual excitations to form the resultant system response. A linear system is, therefore, much easier to analyze than a nonlinear one. Computationally speaking, it is desirable to approximate a nonlinear system by linearizing it under specific, local conditions. For example, nonlinear regression is typically a computationally imposing task in statistics. Fortunately, it can be performed by conditional least-squares linear-regression techniques. This is accomplished by estimating the regression coefficients for a given set of observations, hence the word conditional.

Location factors: In spatial-temporal analysis, we try to discern those factors that have spatial implications. For example, transportation is a significant factor in residential decisions. In these decisions, one trades off housing cost with transportation cost. Together with housing cost, transportation becomes part of the location expenditures in the household budget. This contrasts with non-location expenditures, such as food, clothing and savings, for example.

Macrostate, mesostate, and microstate: A travel pattern can be described conveniently in terms of these three states. Macrostate is the most aggregate description of a travel pattern, while microstate is the most detailed. For example, a macrostate description would only indicate the total number of trips originating or terminating in a zone. In contrast, a microstate description would identify each trip individually about where it is heading. Correspondingly, the former requires the least information and the latter the most. Suppose we wish to characterize the travel pattern in terms of the macro and meso states. The most likely mesostate or macrostate is assumed to be one with the greatest number of possible microstates. Thus we maximize the possible microstates in an aggregate description of travel pattern. (See *entropy maximization*.) In other words, we ask for the least amount of information (*information minimization*) to characterize the travel pattern consistent with some givens, such as the total number of trip originations.

Maximum principle and adjoint equation: In control theory, we optimize the performance of a system by manipulating the control variables and theoretically the state variable over time. The optimization procedure can be executed by first optimizing with respect to the control variable and then the state variables. The first optimization equation is termed the maximum principle and the second the costate or adjoint equation. These two conditions plus the state equation are the necessary conditions for optimality over time.

Median versus antimedian: A median is a location that is the closest to the demands on the average. Thus, a retail chain may wish to open a store close to the population. An antimedian is just the opposite. It puts the facility away from the demands. An example is to locate an airport away from the population for noise considerations. In short, the median problem minimizes the distance to the total regional demand, while the antimedian problem maximizes the distance to the demand. Medianoid refers to a median on a tree (which is a network without any closed loops or cycles).

Medicenter versus anti-medicenter: A medicenter, also known as centian (which stands for *center* and med*ian*), is a hybrid between a median and a center. It takes care of both the proximity to demands as well as the reduction of the most adverse exposure. One can argue this is the best criterion for locating a landfill—close in general but not too close for the most irritated. Anti-medicenter is just the opposite of medicenter. It maximizes the sum of the weighted distance where the demands serve as weights. Yet at the same time, we minimize the maximum weighted distance. It may be the best for locating an airport, which should be a reasonable distance away from the regional population, yet within reach for the most remote residents.

Min-max versus min-sum: There are two traditional criteria in locating facilities. One is the mini-max criterion and the other is the mini-sum criterion. The min-max criterion results in a center, wherein the farthest demand is to be brought as close to the service facility as possible. The min-sum criterion, on the other hand, results in a median, a facility that is as close to the demands as possible on the average. Within these two general criteria, quite a few variations are possible, giving rise to a rich array of facility location models.

Model versus submodel: A model is a mathematical abstraction of a problem. In building large-scale mathematical models, it is often convenient to break down the model into its parts, called submodels. The model is now made up of several submodels. The art of modeling then becomes a matter of how to account for the interaction between these submodels accurately. This to ensure that analyzing each submodel, one at a time, will not lose any property or behavior of the overall model.

Model identification and specification: In fitting an econometric model statistically, there has to be an appropriate match between the available data, the structural equations describing the model, and the corresponding ability to calibrate model coefficients. Classical literature points toward the proper balance between *endogenous* (dependent) and *exogenous* (independent) variables. Otherwise, a model can be overspecified and overfitted. A regression line that is fitted over two data points, for example, is both a mis-specified and overfitted model. In spatial econometric models, we are explaining spatial dependence between the variables defined at various locations. Spatial data come in different levels of aggregation, with some geographic units bigger and others smaller. To calibrate a homogeneous model, we properly define spatially lagged variables with predefined weights. In image processing, we refer to these weights as masks. Through these lagged or weighted variables, a model can be readily calibrated consonant with the proper data format. Here, some lagged variables may become endogenous variables, while others become exogenous variables. Combined with other explicitly given spatial variables, a meaningful econometric model can correspondingly be constructed. Mis-specification of a model can give rise to unreasonable results that may look fine statistically, but has little meaning in modeling the system at hand.

Monocentric: Classic regional-science literature has idealized a typical city as having a single downtown with the highest development density, and the rest of the development thins out toward the fringes. This simple monocentric city form is constructed obviously for convenience. But it yields a number of insights, based on which more complex models can be built.

Monotonic/monotonicity: A monotonic function is either non-increasing or non-decreasing. It has a nice analytic property for a number of spatial-temporal applications. Among these is the economic-activity generation process in a study area, in which the seed of economic development germinates multiplier effects on the local economy. Barring any catastrophic intervention, the resulting population/employment activity level is shown to be non-decreasing. In the absence of bifurcation, the growth stabilizes in time to a limit. This process, and its monotonicity property, forms the basic building block of a surprising number of land use models.

Multi-attribute/multicriteria/multi-objective: Utility theory is the foundation of economics and operations research. The basic premise is that a number of disparate metrics can be translated into a common unit called utiles. Once this is done, cross comparison can then be made among alternatives with seemingly incommensurate attributes or criteria. This is generally accomplished by a multi-attribute utility function, which combines the incommensurate attributes or criteria through weights and scaling constants. Cross comparison among alternatives can still be possible without a multi-attribute utility function, although in a more limited sense. For example, a shirt that is cheaper and better quality is always preferred to one that is more expensive and inferior in quality. Here, no utility function needs to be constructed to combine price and quality, the two different attributes, into utiles before a decision can be made between them.

Multicommodity or multiproduct: Rather than monolithic, often one differentiates the type of service provided, or the purpose of tripmaking. Multicommodity or multiproduct flow results in such a situation, with each type of service or trip tagged. The analyst needs to decide the most parsimonious model commensurate with the problem at hand, wherein the complexity is justifiable on the grounds of model realism. In many cases, the multicommodity or multiproduct model is a simple extension of the single commodity/product case, at least mathematically speaking.

Multinomial logit model: Often we wish to classify entities into multiple groups based on their attributes. A statistical model is often formulated with a response variable having two or more categories. In the case of two responses, the model is called binomial, and with three or more responses, it is called multinomial. An example is to find a neighborhood in which one locates a home. The multinomial logit model is a common way to do this and over recent years has found its way into the location and transportation literature. In many ways, it is related to the venerable gravity model that is pervasive among those involved in regional science. Instead of using power functions to describe accessibility, the logit model prefers exponential utility functions. Among the advantages of the logit model is that it is based on some widely accepted behavioral assumptions regarding the utility associated with belonging to each group, such as the accessibility to various opportunities in the study area should one locate a home in a particular neighborhood. The logarithm of the model is linear, making it convenient for calibration using ordinary-least-squares regression.

Multispectral sensors: Today's remote sensing devices, such as satellites, are equipped with more than one sensor. Several sensors are used to collect different *emittance* wavelengths, resulting in a signature of an object being observed as characterized by the different waves the object emits. A much more positive identification of the object can be obtained this way compared with a single sensor that captures only one type of wavelength. For example, the human eye is a sensor that is limited to see the visual wavelengths, which is but a minute fraction of the signals emitted from an object. For this simple reason, multispectral sensors can literally see the invisible.

Non-dominated, efficient, or Pareto optimal solutions versus supremum: A cornerstone of multicriteria optimization is the concept of dominance. Thus site A, which is cheaper and more functional, is a better site than B which is more costly and less functional. Here, A is the non-dominated solution or the Pareto optimum, and B is the dominated one. This idea can be easily generalized to many alternative sites, as long as we compare only two at a time. After an exhaustive comparison between all pairs and discarding all dominated alternatives, those remain form the non-dominated, efficient, or Pareto-optimal solution set. In contrast, the *supremum* of a function refers to either the maximum or the minimum on a unidimensional scale.

NP, NP-complete: *NP* stands for non-deterministic polynomial, characterizing problems that have not been shown to be solvable within execution time that goes up polynomially with the size of the problem. *NP*-complete (*NPC*) problems constitute a subset of *NP* problems. The implication is that once a member of the *NPC* class of problem is solvable within polynomial time, the entire class of problem will also be solvable within polynomial time. Being an integer program, discrete facility-location models are at best an *NPC* problem, making it a difficult problem to solve.

Object-oriented programming: One can think of solution algorithms for a mathematical model as a set of computational procedures to process a set of input data, resulting in a set of output data. The solution algorithm is as efficient as how fast one can process the data. It follows that when the data are organized in the right format, they can be processed faster than otherwise. Efficiency can also be achieved if a set of computational procedures can be used time and again for a number of purposes. This avoids coding a separate routine for each application. Object-oriented programming is one good way to accomplish these objectives. In location-allocation models, this means preprocessing of inter-point distance data as both candidate and demand strings, which serves to update an allocation table. In a data transfer protocol for geographic information systems, this means that we define precisely such objects as a node/vertex and a chain in a vector data structure. Specifically, they are stored in relation to other related node/vertex-chain information. In this way, the efficient transfer of complete chains is facilitated.

Objective function or functional: In optimization problems, a figure of merit is usually maximized or minimized. For example, profit is to be maximized while cost is to be minimized. This figure of merit is expressed in terms of an objective function or functional. Thus profit or cost is expressed in terms of a set of decision variables or control variables respectively. The two terms—function and functional—are traditionally used in different disciplines, but are in fact equivalent. Both define a domain whose elements are functions, sets, and the like. According to traditional usage, objective functionals (the integration of a functional overtime) normally have a time element associated with them, while objective functions are generally static expressions.

Optimality and stationarity: A function satisfies its optimality conditions when it is maximized or minimized at a point within the feasible region. A continuously differentiable function is optimized when it has a relative maximum or minimum at a point assuming a stationary value. This value again lies at an interior point of the feasible region. At this point, the function is said to be stationary. A stationary point is obtained by setting the gradient of the functional to zero. For this reason, it also includes an *inflexion point* of the function, which is not a local optimum.

Orthogonal/orthogonality: Independence among attributes is necessary for the construction of a meaningful multi-attribute utility function, since we will not be double counting an attribute. If two attributes are independent, they are also orthogonal. Orthogonality is a more general term than independence, however, since there are several types of independence in multi-attribute utility theory, while orthogonality is pretty much a monolithic concept.

Orthonormal vectors: If a set of vectors is orthogonal and normalized, the vectors are said to be orthonormal. These vectors form a convenient algebraic basis for referencing. For example, in a stochastic compartmental model, the transition rate space can be conveniently characterized by a set of orthonormal vectors. These vectors map out the possible transitions from the current *j*th compartment (state) to other compartments (states). Thus the transition rate to a neighboring compartment is changed by a unit increase or decrease of activity level or price. Here the increase or decrease is implemented by adding or subtracting a unit vector from the current activity or price level.

Parametric versus non-parametric statistics: Means and variances are typical ways to summarize statistical information. The use of the parameters such as means and variances is a good example of parametric statistics. Data description can take on other forms, however. In a very small sample, means and variances are no longer meaningful, since there are simply too few data points for these parameters to become representative of the entire data set. Dispensing with the use of parameters, non-parametric statistics serves to characterize the data under these circumstances. An example of non-parametric statistics is entropy, defined here as the various representations of the data permissible within some givens. In the example of a small data set, the givens may be the precious few observed values of the data. There are obviously quite a few underlying data populations that could manifest themselves in these observed values. The number of possible underlying data sets in this case is called *entropy*. One normally asks for the maximum number of characterizations of a data pattern that requires the least amount of information *(minimal information)*, or entropy maximization. This means we seek the largest possible number of underlying data populations that are consistent with the few observed values. Notice here that not only is parameter estimation not required, no knowledge of the underlying data distribution is necessary. Non-parametric statistics plays a significant role in spatial statistics—statistics that arise in facility location and land use.

Parameterization: In spatial allocation models such as the gravity model, a key term is the *accessibility* factor, defined roughly as the inverse function of distance. It turns out that the exponent associated with spatial separation is a critical parameter. It determines the importance of interaction between origins and destinations vis-a-vis the dispersion or the distribution of activities such as population and employment among neighbors. As it turns out, when this exponent is infinite, there is nothing but interaction, or the assignment of supplies to demands. When the exponent is very small, there is predominantly continuous allocation of activities among its neighbors. One can say that this exponent is the key to characterizing a model as discrete facility location or continuous activity allocation (land use) models. One of the aims of this book is to show that through the transformation of distance measures, one can relate apparently different spatial-temporal models to one another. Another parameterization example is the relationship between a median model and a center model in facility location. It has been established that upon appropriate transformation of the spatial-separation function through the exponent, a center model can be reduced to a median model. There are several other examples, but these two cases serve as graphic illustrations. (See also exponentiation above.) In time series, parameterization means specifying the degree of differencing and the number of time lags built into the data, and so forth. It characterizes the time series.

Pluralistic/pluralism: One of the challenges of facility location and land use is the multiplicity of viewpoints held by a diversity of stakeholders. Often citizens have a viewpoint opposite from that of the local government, and industries have different objectives from environmentalists. Analysis techniques need to explicitly recognize this pluralism and produce useful information for all stakeholders in the decision-making process.

Polyhedron and polytope: Many discrete facility-location problems are solved by mathematical programs. The simplest mathematical program is linear programming, which can be solved readily by off-the-shelf software. These programs work on the principle of searching among the faces of a polyhedron, or a many-sided multi-dimensional body defined by the linear constraint inequalities of the linear program. A polytope is simply a bounded (or finite) polyhedron. It can be proved that one only needs to examine the extreme points or edges of a polyhedron for an optimal solution, where an extreme point or edge is the place where two or more faces come together. (See Appendix 4 [Optimization]; see also *extremal conditions*.) The same concept can be carried over to other types of mathematical programs, such as integer programs and nonlinear programs, except the search for optimality becomes much more complex.

Queuing: In this book, service vehicles are often lined up at the depot to respond to calls or demands during busy periods. Until a vehicle has finished servicing a demand, it cannot be dispatched to another demand location. In this case, one has to wait for a vehicle to become available; the vehicle then takes time to travel to the scene; it spends time servicing the demand; and finally returns to the depot ready for assignment again. Typical queuing literature, or the study of waiting lines, is now extended to a spatial context, concomitant with the greatly expanded analytical complexity.

Recursive operation, recursion, and recursive programming: In the temporal dimension of spatial-temporal analysis, one wishes to lay out the evolution of development from one time stage to another. To design the most desirable plan in the long run, the question is whether it is sufficient to do the right thing at each stage. Irrespective of whether it is or not, one can only execute a local decision at each stage, and he or she does it repeatedly for each time stage. We call this repetitive process recursion. The same idea applies to stagewise decisions in which only the spatial dimension is involved. An example is planning airline flights. A corporate planner starts out with candidate nonstop flights, then configures a one-stop flight made up of a new leg attached to an existing nonstop, and finally configures a two-stop flight by adding yet another leg to a one-stop. Again, the overall decision is broken down into a series of recursive decisions. Many computations are recursive in nature, including the filtered and one-step ahead recursions in adjusting a time series to a new pattern. When successive optimal recursions result in an overall optimum, we are dealing with a Markovian process. When global optimality is not guaranteed, we are merely dealing with a recursive program.

Satisficing: There are two types of achievements. The first is "the more the merrier." An obvious example is money; few would argue against having more money, and the more the better. The second achievement is more precise.

One aims to obtain a specific amount of an item, such as "achieving the clean air standard." When a standard or threshold is achieved, we have obtained a satisficing solution. The concept of satisficing is therefore related to a threshold. Once a threshold is exceeded, there is no preference between the resulting solutions, no matter whether one barely exceeds it or exceeds it by a large margin. Related to these concepts is Goal Programming, where the deviation from a preset goal is to be minimized. A goal is defined as a target for achievement, such as an artist who mixes colors in a palette to achieve an intended color the artist has in mind. Oftentimes, it boils down to coming as close as possible to the color in the artist's mind. The unwanted deviations can be ordered into priority levels. Minimizing a deviation in a higher priority level is infinitely more important than any deviations in lower priority levels. This is known as Lexicographic or Pre-emptive goal programming. Considering the three fundamental colors—red, blue and yellow—the artist may value coming closest to the red hue more than the yellow tone, and she values the yellow tone more than the blue tint.

Scaling and re-scaling: In spatial allocation of activities, it is necessary to ensure the sum of the zonal allocations add up to the grand total for the region. As one derives population from employment and vice versa, the sum of the derived zonal allocations does not necessarily agree with the exogenously forecast regional total. A scale factor simply ensures this happens. Another example of scaling is the calibration of a multi-attribute utility function. Questioning the decision maker will yield a set of weights among the criteria. But there is no guarantee that the utility function so obtained will be 0–1 ranged, the convention for utility functions. A scale factor simply makes it happen.

Single versus multiple periods: Oftentimes, it is important to consider the expansion of a facility or facilities over time. When a facility is treated as a discrete entity with a location and a service capacity, this multiperiod expansion problem is anything but trivial. If we expand from existing facilities already in place, we may not be able to do as well as starting with a clean slate every period. The goal of facility planning is to provide such continuity between single-period decisions and to come out with the most desirable evolution over time. Again, while an aggregate statement of such a problem, such as in terms of total service capacity to be provided, is relatively straightforward, the spatial statement of the problem compounds the complexity rapidly.

Single versus multiple products: A facility can provide only one type of service or that it can provide different ones. Closely related to this is the hierarchy of service provision. For example, a facility can provide up to a certain level of service and no more, or it can provide all types of services. Once distinction is made regarding the types of services, only an appropriate facility can render a particular service. At the same time, more than one capable facility may cater to the needs of a demand or customer. Facility B can serve as a backup in case facility A can no longer deliver the demand, or both A and B can satisfy the total needs of a customer together. This tremendously complicates the one service provider for one customer paradigm. This idea is not new in urban land use. There we have the equivalent concept of trip purpose, or the type of trip that is being executed. For example, office buildings take work trips while parks and recreational facilities take nonwork trips. They provide different products or services. We find mostly work trips during peak hours of the day and nonwork trips during off peaks. Among recreational facilities are state parks, movie theaters, and bowling alleys that can offer an alternative form of entertainment should a particular recreational facility become unavailable. For example, the seat capacity at a theater may dictate a substitute, or backup, recreational alternative, either a theater at a different location or perhaps a bowling alley.

Siting: In this book, the term siting is used interchangeably with facility location. With only minor exceptions, we do not deal with site layout in detail (although the theory is similar). Rather, we are concerned with the location of the facility in relation to other facilities and the aggregate design parameters of the facility, such as size and capacity.

Software or computer programs: In today's analysis world, seldom does the analyst perform tasks without the aid of computer programs or software. There really is no centralized software package for facility location and land use to date. The compact disk that comes with this book is only a sample of what would eventually be a generalized software suite for this field. Such a suite may be similar to the office suites for various office functions offered with today's personal computers. Clearly the eventual package should contain quite a few elements, including remote sensing/geographic information system, location-allocation procedures, location-routing algorithms, and land-use forecasting tools. Supporting or utility routines should include optimization procedures (such as CPLEX), statistical routines (such as SAS), and stochastic/simulation programs. Part of the aim of this volume is to provide the interested readers with food for thought for the design of an eventual facility location/land use software package.

Source and sink: In the location literature, sources are equated with a service or production facility, while sinks are demand locations, where the customers are located. Thus in both discrete and continuous (planar) location problems, they are the places where service or commodity flows originate and terminate respectively. This concept comes naturally with the discrete network-flow literature, which traditionally has similar terminologies in place. In continuous problems, sources and sinks are among several stable or unstable fixed points, singularity points, or equilibrium points. These points in general characterize the flow patterns on a plane. Existence of these points introduces discreteness of facility and demand locations in an otherwise homogeneous pattern. This lessens the idealized distinction between discrete and continuous location problems.

Spatial versus aspatial analysis: A distinguishing feature of this book is that it explicitly considers geographic attributes, network effects, and interaction between economic activities among different areas within a region. In other words, it analyzes problems with full recognition of the spatial dimensions. In contrast, aspatial analysis deals only with aggregate attributes such as the population and employment in the entire region, the total amount of retail floor space, the total acreage of parks and recreation areas, the total number of hospitals, and perhaps their growth over time. It does not disaggregate by zones or other subareal units, neither does it deal with interzonal interactions such as commuting between employment centers and population centers. Naturally, spatial analysis is much more complex that aspatial analysis. (See also *area* and *subarea*.)

Spatial dependence and independence: When a spatial unit influences or is being influenced by its neighbors, the subject unit is said to be spatially dependent on its neighbors and vice versa. On the other hand, if all spatial units are truly random, they are said to be independent of one another. In this case, the assumed value of a spatial unit *i* has no relationship to the value of unit *j*. Spatial dependence is usually expressed in terms of weights between units *i* and *j*, where a larger weight connotes a heavier dependence. A totally independent set of spatial units is referred to as a *random field*.

Stability and instability: A key property of a dynamical system is structural stability or instability. Inherent in the system is the innate ability to return to equilibrium after perturbation or that it transforms into disequilibrium. We refer to the former as a stable system and the latter unstable. In terms of spatial structures, the flow pattern of services and commodities between facilities and demands is a result of the economy that governs the study area. Locations of these facilities can either be stable or unstable. Example of a stable facility location is a *source* (where flows originate naturally). Certain saddles, with the associated *separatrices* or the dividing flow lines separating flows so that they are not converging on the saddle point, are unstable. Flows can be *periodic* or cyclic. In this case, a family of concentric circles around a fixed center suggests instability. On the other hand, a limit cycle—in spite of slight irregularities in its orbit—will always return to its starting point and hence it is stable. (See also equilibrium and disequilibrium.)

Statics versus dynamics: When a phenomenon is the same irrespective of time, it is said to be static. By contrast, a dynamic phenomenon changes over time. In this book, traditional facility-location models are often static in nature. Recent advances in stochastic facility-location models have extended the horizon to time-varying location decisions. By contrast, land use models are often used to forecast population and employment for a target date. In that light, they are dynamic in nature as one forecasts iteratively over, say five-year increments into the future.

Total unimodularity: If the constraint matrix of a linear program (LP) is totally unimodular, and the right-hand side is an integer vector, the LP will yield integer solutions. A network LP has exactly such property. To the extent that a facility location problem can often be formulated as a network LP, the desired integrality property is most valuable in solution procedures.

Trip and route, versus tour: Much of a location decision is the result of considering accessibility to economic, social, and recreational opportunities. To reach these opportunities, trips may have to be executed either by the population or the provider. In this case, the population follows a route to the goods and services, or the goods or services have to be routed by the provider to the population. When a special delivery is made by the provider, the vehicle used for the delivery may be productive only in one direction, namely on the way to the demands when carrying the goods and services. The return trip is often empty and not productive, unless another load is backhauled to the provider. On the other hand, if the population combines several errands in a trip, these errands can be completed in a "round robin" visit to several service providers. We call this a tour, which can likewise be executed by the provider to deliver the goods or services.

Unimodal, bimodal, or multimodal: In a frequency distribution, there may be only one single peak. We call this distribution unimodal. When there are two peaks, it is bimodal. In general, there can be multiple peaks, constituting a multimodal frequency distribution.

Univariate, bivariate, and multivariate models: For pedagogic reasons, most subjects are introduced in its simplest form, often involving one single variable. Multivariate models, however, are common in spatial-temporal analysis since each unit (for instance, a zone in a region or pixels in a photo) is usually represented by separate variables. As a result, there are many entities to analyze. Bivariate models are often used as a transition from univariate to the more complex multivariate case.

Univariate spatial time-series: A time series is a sequence of observations on a single or multiple variables. This book is particularly interested in the latter when several spatially related variables are examined. It is often of interest to analyze the underlying pattern of such a time series, so that one can forecast future trends. One such analysis is to regress the time series with its lagged series, and the quality of such a model is measured by a goodness-of-fit parameter such as *autocorrelation,* the temporal counterpart of regular Pearson correlation in classical statistics. When each of these spatial variables is expected to behave similarly, one can simply construct a univariate time series. Once calibrated, such a time series would describe every spatial variable equally well over time. One pattern that exists in such a time series is seasonality. For example, people travel more in the summer months than winter months annually. While *differencing* may remove seasonality from a time series, a seasonal pattern may still remain in stationary data. This poses an additional challenge in model identification. Not only does one need to identify the lags for the autoregressive and moving average components (usually denoted by p and q respectively), additional specifications on the season length for the two components (s_p and s_q) need to be made. These two tasks, identification of p/q and s_p $/s_q$, are performed in the sequence as stated. There is a close parallel between the identification of a seasonal non-spatial model and the identification of a univariate spatial time-series. Normally, the spatial time series can be analyzed as a scalar sequence of observations that in all appearances, resemble a seasonal time series. One can employ steps similar to the seasonalized procedure to identify a spatial-temporal model.

Utiles: Utiles are the common currency of exchange among incommensurate quantities. Through the construction of a multi-attribute utility/value function, for example, one can combine apples and oranges together in a common unit called fruit. The common unit allows cross comparison to be made among two very different quantities, including tradeoffs among them. Utiles is the basic building block among most operations research and *econometric* analyses.

Variance, covariance, correlation, and autocorrelation: Variance (or standard deviation) of a single random variable is the spread of the data around the mean. When two or more variables are involved, the metric is broadened to measure the scatter of data around the trend line explaining the relationship between the two (or a dependent and several independent) variables. The less the scatter, the more the variables are correlated via a trend line (surface). The more the scatter, the more questionable the correlation. Formally, covariance between a pair of variables is the product of the standard deviations of the two given variables and the correlation between them. Thus one can see it reduces to the variance of a single variable when the two random variables are identical, or when the correlation between the two is unity. Where the concept is carried over to the spatial (multivariate) and temporal dimension, we have variance-covariance matrix and autocorrelation. A variance-covariance matrix has the variances along its diagonal for the same variable and covariances at off diagonal elements for a pair of different variables. Autocorrelations are correlations between a time series and itself shifted by a certain period of time, the former series forms the dependent variable and the latter the independent variable. In analyzing time series, the use of auto-covariance and auto-correlation functions aids in the identification and estimation of the models.

Vector versus raster spatial-data storage: Spatial data can be stored in two generic formats: vector and raster. The former is the traditional way, advanced long before the digital computer and satellite images. It exploits relations among points, lines, and areas. It has a more compact storage requirement and can be very precise in selected applications. However, the latter, because of its grid or lattice structure, has the distinct advantage of format uniformity when various data sources are merged, as long as the data are discretized into a grid. It is also amenable to a wide variety of image restoration and enhancement routines. Many of today's spatial data are digitized for storage, precisely for these reasons. Obviously, the problem drives the logical format for data storage and no single format is inherently superior to another. The data stored in vector or raster format can be socioeconomic attributes such as population and employment, or they can be gray values in a panchromatic image. For uniformity, we choose to use the generic term activity in this book to describe any spatial data value.

Weighting: In evaluating alternatives, it is desirable to combine several criteria or attributes into a single metric called *utiles*. In doing so, it is common to weigh each criterion/attribute differently according to its importance and then add them together, by means of a weighted sum for example. Obviously, the overall utile of an alternative is different depending on the specification of weights. The ranking among alternatives according to utile is therefore different depending on the assignment of weights.

Work versus non-work trips: Transportation planning is a major factor in land development. In transportation, distinction is made between trips made to employment location and trips for other purposes. Generally speaking, work trips are inelastic, while nonwork trips are much more elastic and are often discretionary. Work trips determine factory and other employment locations with respect to residential locations. Nonwork trips, on the other hand, determine the siting of shopping malls and other services, again vis-a-vis residential neighborhoods.

Appendix 6

Abbreviation and Mathematical Symbols

This book aims to serve those who analyze spatial-temporal information. In this appendix, we put the common abbreviations in one place for easy reference. Compiled here is an illustrative list of acronyms used in the field, defined here as the beginning initials of a technical term. Also included are "alphabet soup" terms, which are the abbreviated names in common usage that do not necessarily correspond to the beginning initials. By intent, the list goes beyond those terms used immediately in this text. Similar to Appendix 5, it serves as a quick, easy recreance for those who might otherwise be "intimidated" by the profusion of technical jargons in the literature.

Also included is a comprehensive list of mathematical symbols. Again, the list goes beyond this immediate text. It includes those that appear in the companion volume: Chan, Y. (2005). *Location, transport and land-use: Modelling spatial-temporal information*. Berlin and New York: Springer. The mathematical symbols are totally consistent between this volume and the accompanying volume.

2SLS	two-stage least squares (calibration procedure)
ACF	autocorrelation function
ADBASE	a multicriteria linear-programming model code
AHP	analytic hierarchy process
AIC	Akaike information criterion
AMDAHL	type of main frame computer
AMPL	modeling programming language
ANOVA	analysis of variance
API	application programming interface
ARC/INFO	a geographic information system
ARIMA	auto regressive integrated moving average (model)
ARMA	auto regressive moving average (model)
ASSIST	Ambulance System Site Inspection Simulation
ATLAS	a geographic information system
AVHRR	advanced very high resolution radiometer
B & B	branch and bound
BFS	basic feasible solution
BW	Benabdallah and Wright (algorithm for districting)
CAD	computer-aided design
CBA	capacitated basic algorithm
CD-ROM	compact disk-read only memory
CDF	cumulative density function
CFLOS	cloud-free line of sight
CGI	Common Gateway Interface
CI	consistency index
CLI	Command Line Interface
CLUG	Community Land Use Game

CPLEX	a linear and integer programming code
CRPC	Centre Region Planning Commission
CS	Central store
CSPE	classical spatial price equilibrium
CW	Clarke-Wright (routing heuristic)
D-A	digital-analog
DCPLP	dynamic capacitated plant-location-problem
DCS	Defense Courier Service
DEA	data envelopment analysis
DIME	dual independence map encoding
DLG	digital line graph
DLG-E	digital line graph-enhanced
DM	decision maker
DMU	decision-making unit (in data envelopment analysis)
DN	digital number (of a pixel)
dof	degree of freedom
DP	dynamic programming
DPM	downtown people mover
ELECTRE	a discrete-alternative multicriteria-optimization software
EMPIRIC	a linear econometric land-use model
EMS	Emergency Medical Services
ETAC	Environmental Technical Applications Center
FANAL	Factor analysis program in the EMPIRIC model
FFT	fast Fourier transform
FGDC	Federal Geographic Data Committee
FI	full industries
FIFO	first-in-first-out (queuing discipline)
FIPS	Federal Information Processing Standard
FORCST	Forecast program in the EMPIRIC model
FSCORE	Factor Scores program in the EMPIRIC model
F-W	Frank-Wolfe (method)
GAMS	Generalized Algebraic Modeling System
GASP	General Activity Simulation Program
GBF	geographic base file
GEODSS	ground-based electro-optical deep-space surveillance
GIS	geographic information systems
GLONASS	navigation satellite system operated by the Commonwealth of Independent States
GMI	Gray-McCrary index
GMP	generalized median problem
GNSS	Global Navigation Satellite Systems
GOES	Geostationary Operational Environmental Satellites
GPS	Global Positioning System
GPSS	General Purpose Simulation System
GRASS	Geographical Resource Analysis Support System
GS	goal setting
GSARP	generalized search-and-rescue problem
GUF	group utility/valve function
HFDF	high frequency direction finder
IBIS	image based information system

IC	information criterion (decision rule)
ICM	iterative conditional mode (algorithm)
IGDS	a geographic information system
IGUF	individual group utility function
IMSL	International Mathematical and Statistical Library
INFORMAP	a geographic information system
IOM	intervening opportunity model
IP	integer programming
KKT	Karash-Kuhn-Tucker (condition)
LANDSAT	an earth surveillance satellite
LD	Lagrangian dual
LGPL	Lesser General Public License
LP	linear program, linear programming
LR	Lagrangian relaxation
LRP	location-routing problem
LS	Local store
MADA	multi-attribute decision analysis
MARMA	multivariate auto-regressive moving-average model
MAUT	multiattribute utility theory
MCDM	multicriteria decision making
MCLP, MLP	multicriteria linear program
MCO	multicriteria optimization
MC-SIMPLEX	multicriteria simplex
MCSLP	maximum consumers'-surplus location problem
MDMTSFLP	multi-depot multi-traveling-salesmen facility-location problem
MDP	Markovian decision process
MDVRP	multi-depot vehicle-routing problem
MICROSOLVE	an operations-research software suite
MIP	mixed integer programming
MIP83/XA	a linear and integer programming software
MNBLP	maximum net-benefit location problem
MOLIP	multiple objective linear integer program
MPSX	Mathematical Programming System extended
MSF	minimum spanning forest
MSFC	multiple space filling curve
MSS	multispectral scanner
MTC	marginal transportation (economic) cost
MTSFLP	multiple-traveling-salesmen facility-location problem
MTSP	multiple traveling-salesmen problem
MULSTARMA	multivariate spatial-temporal auto-regressive moving-average model
NASA	National Aeronautics and Space Administration
NDCDB	National Digital Cartographic Database
NETSIDE	a network-with-side-constraints software
NIMBY	not-in-my-backyard (syndrome)
NLIP	nonlinear integer program
NLP	Nonlinear programming
NOAA-n	National Oceanic and Atmospheric Administration *n*-series (meteorological satellites)
NP	non-deterministic polynomial

NPC	*NP*-complete
NSC	network-with-side-constraints
NVI	normalized vegetation index
NWPA	Nuclear Waste Policy Act
NWS	National Weather Service
OBE	operating basic earthquake
OGC	Open Geospatial Consortium
OLS	ordinary least squares
P	polynomial
PACF	partial autocorrelation function
PAR	calibration parameters of the auto-regressive terms in an ARMA model
PC	personal computer
PDF	probability density function
PI	partial industries
PMA	calibration parameters of the moving-average terms in an ARMA model
PMT	person miles of travel
PMTSFLP	probabilistic multiple-traveling-salesmen facility-location problem
PMTSP	probabilistic multiple-traveling-salesmen problem
POLYMETRIC	a nonlinear econometric model
PROC	procedure in the SAS software
PRT	personal rapid transit
PTSFLP	probabilistic traveling-salesman facility-location problem
PTSP	probabilistic traveling-salesman problem
PTST	probabilistic traveling-salesman tour
PVRL	probabilistic vehicle-routing location
QAP	quadratic assignment problem
R2	Variance
RADARSAT	Canadian satellite with all weather and night-time capability
RP	recursive programming
RIOM	regional input-output model
RISE	route improvement synthesis and evaluation (algorithm)
SAR	seasonal auto-regressive (model); search and rescue
SARMA	seasonal auto-regressive moving-average (model)
SAS/OR	an operations-research software suite within the SAS business analytics software
SCA	an integrated time-series-analysis computer-program
SDTS	spatial data transfer standard
SEE	standard error of estimate
SFC	space filling curve (heuristic)
SIMAN	a discrete-event simulation language
SIMSCRIPT	a discrete-event simulation language
SIPs	Statistically Improbable Phrases
SIR	spaceborne imaging radar
SLAM	a discrete-event simulation language
SMA	seasonal moving average (model)
SMSA	Standard Metropolitan Statistical Area
SPANS	a geographic information system

SPE	spatial price equilibrium
SPLP	simple plant-location problem
SPOT	a French commercial satellite
SPSS	a statistical software system
SSM	subregional simulation model
SSR	sum of squared residuals
SSTMA	seasonal spatial-temporal moving-average (model)
STACF	spatial-temporal autocorrelation function
STATESPACE	procedure within the SAS software
STARMA	spatial-temporal auto-regressive moving-average (model)
STMA	spatial-temporal moving-average (model)
STPACF	spatial-temporal partial-autocorrelation function
SYSNLIN	procedure within SAS for vector-time-series analysis
SYSTAT	a statistical software
TAZ	transportation analysis zone
TCTSP	time-constrained traveling-salesman problem
TCVRP	time-constrained vehicle-routing problem
TIGER	Topologically Integrated Geographic Encoding and Referencing (system)
TM	Thematic Mapper
TRANSCAD	a geographic information system for transportation applications
TS-IP	Training System-Image Processing
TSFLP	traveling-salesman facility-location problem
TSP	traveling salesman problem
TST	traveling salesman tour
TUM	totally unimodular
TVP	topological vector profile
UNEP	United Nations Environment Programme
USGS	United States Geological Survey
USPE	univariate stochastic model preliminary estimation (program)
VARMA	vector auto-regressive moving-average (model)
VBA	Visual Basic for Applications
VGA	video graphics adapter
VI	vegetation index
VRP	vehicle routing problem
WMTS-1	Wisconsin Multiple Time Series (program)-1st edition
X-SAR	X-band Synthetic Aperture Radar
ZOOM	Zero-One Optimization Model

List of Symbols

a A calibration constant; for example, it is the service-employment multiplier or population-serving ratio (number of service jobs generated from one household or resident)

$\tilde{a}$ Intercept regression-coefficient as a random variable

$\tilde{a}^*$ Specific value of $\tilde{a}$ corresponding to a sample of data points

a' Acceleration of a vehicle; also a constant parameter, such as unit cost of commuting (cost per unit-of-distance travelled), or the exponent of the development opportunity W_j at destination j

a_i Calibration parameter corresponding to the utility increase in zone i, where utility is some measure of composite accessibility to the zone; also the population-serving ratio at zone i

a_t Estimation-error or noise term for a series of data ($t = 1, 2, \ldots$) usually in a 'normalized' time-series, or after the data have been differenced to a stationary series; the estimated error or noise in Kalman filtering; also referred to as innovations when it is white noise

a_i' Physical area of geographic sub-unit i or the demand-generating potential of i

a_t' Measurement error in a Kalman-filter time-series, representing the difference between observed and measured data

a_D Error term in a demand econometric-equation

a_S Error term in a supply econometric-equation

a^W Weighted labor-force-participation-rate, where the weights are the percentages of regional population in each zone

a^p The pth-sector employment-growth-rate in the entire study-area

a_{ij} Parallel to its single-dimension analogue, a_{ij} is an error- or noise-term in the spatial context; it has a zero mean and a constant variance; also stands for the entries in the $\bar{\mathbf{A}}$ matrix

a_i^u Convex combination of the population-serving ratios, with normalized accessibilities to zone i as weights

a_j^p Employment multiplier considering the population-serving ratio, i.e., $(1 + a_j)$—segregated both by economic-sector p and by zone j here

a^{kl} Calibration parameter in a predictor-prey equation-set showing the interaction between the kth and lth species

a_{kl}' The kth output (benefit) measures due to decision-making-unit j considering both nonspatial and spatial attributes (see also $\bar{\mathbf{A}} = [a_{ij}]$)

$a_{ij}^{pq}(k)$ Impact of the pth-state-variable-in-zone-i-at-time-k on the *qth*-state-variable-in-zone-j-at-time-$k + 1$

a'' Threshold for a high-pass noise-filter

$\mathbf{a} = (\leftarrow a_i \rightarrow)^T$ Vector of calibration coefficients in the second stage of 2-stage least-squares, consisting of q entries; also stands for the vector of the error (noise) terms in a spatial-temporal forecasting-model

$\mathbf{a}'$ Vector whose ith element is the ratio of the-household-income to the gross-output-in-the-ith-industrial-sector

$\tilde{\mathbf{a}}$	Interim error-vector or noise-term in a more efficient calibration-procedure for STARMA
$\mathbf{a}_{ij} = (\leftarrow a_{ij}^{k} \rightarrow)^{T}$	Each entry of the $\bar{\mathbf{A}} = [a_{ij}]$ payoff-matrix is replaced by a vector in a linear program, mainly to facilitate a multicriteria, two-person, zero-sum, non-cooperative game; here k is the index for a criterion
α	Calibration constant, or step size in "hill-climbing" algorithms; also the tail of a distribution
α'	Angle between two criterion-functions in multicriteria linear-programming; also a calibration constant
α''	Resulting problem-type after the original problem has been polynomially reduced
$\mathrm{\alpha}_t$	Random-shock or white-noise input-time-series in a transfer-function model
$\mathrm{\alpha}_{ji}^{qp}$	Exponent in a Cobb–Douglas production-function corresponding to the factor input x_{ji}^{qp}
A	Accessibility expenditure for a household (part of locational expenditure); also the area
$\mathrm{A}(\cdot)$	Area of $\cdot$
A_i	Weighted labor-force participation-rate, with accessibility from zone i as weights
A_j	Gross acreage of subarea j
A_j'	Useable gross-acreage of subarea j
A_t	Error term in a "raw-data" time-series
$\bar{\mathbf{A}}_j$	Developable acreage in subarea j
A^B	Basic land-use (A_j^B is basic land-use in zone j)
A^R	Retail land (A_j^R is retail land in zone j)
A^U	Unusable land (A_j^U is unusable land in zone j)
$\underline{A}$	Set of arcs in a network
$\hat{\mathrm{A}}_j^k$	Net acreage in subarea j devoted to the kth land-use
$\mathrm{A} = (\leftarrow A_t \rightarrow)^T$	Vector of disturbance or error terms in econometric or spatial time-series models, consisting of n observations; in 2-stage least-squares, it consists of q entries, where q is the number of endogenous variables
$\mathbf{A}$	As a matrix (instead of a vector), **A** stands for node-arc incidence-matrix in network-flow programming
$\mathbf{A}' = [A_{ij}']$	An $n \times n$ square matrix; for a compartmental model, it is the rate-of-change matrix; and for the matrix of secondary (retail)-employment it is the distribution-rate by zone, where $n = n'$.
$\mathrm{A}_0(t)$	Vector showing rate-of-change with the "outside world" over time
$\mathbf{A}'' = [(i, j)]$	Contiguity matrix with nonzero arc-entries where i is incident upon j
$\hat{\mathbf{A}}$	An $n \times n$ matrix, which converts value-added output vector by industrial sectors to the same vector measured in labor-force base
$\mathbf{A}_j$	Vector of socioeconomic variables at location j, representing such activities as population and employment
$\mathbf{A}(j)$	Column vector in the network-simplex tableau for arc j
$\bar{\mathbf{A}} = [a_{ij}]$	Coefficient matrix of linear-programming constraints, where a_{ij} expresses the incidence relationship between row i and column j;

an example is the *k*th output measures due to decision-making-unit *j*, a_{kj}, in a data-envelopment analysis.

$\mathbf{A}_B$ Basis of a linear program

$\mathbf{A}_N$ Nonbasic part of the tableau in a linear program

$\mathbf{A}^1$ The complicated set of constraints in a mixed integer-program

$\mathbf{A}^2$ The straightforward set of constraints in a mixed integer-program

b Generally a constant parameter, denoting a growth rate, intercept or slope in a linear equation, or the positive exponent of a spatial cost-function etc.

$\tilde{\boldsymbol{b}}$ "Slope" regression-coefficient as a random variable

$\tilde{\boldsymbol{b}}^*$ Specific value of $\tilde{\boldsymbol{b}}$ for a sample of data points

b^U Household budget

b_j The fixed cost of siting a depot at node *j*

b^j Travel-cost elasticity for activity *j*

$b^k(m)$ A scale factor used to adjust the *k*th zonal-retail-employment from one loop of the Lowry model *m* to another $m+1$, where $m = 1, 2, \ldots$

b_{ki}, b_{ik} Slack-flow capacity on slack arc (*k*, *i*) or (*i*, *k*); also the benefit variable in data-envelopment analysis, denoting the weight placed on the *k*th benefit of the *i*th alternative

b_{kji} Benefit variable used in the combined data-envelopment-analysis-and-location model, showing the relative importance of assigning the *k*th benefit to the demand-facility pair *ij*

$\mathbf{b} = (\leftarrow b'_i \rightarrow)^T$ Vector of estimated parameters in ordinary least-squares regression or other calibration procedures, consisting of $k+1$ parameters (including the "intercept"); also the right-hand-side of a linear or mixed integer program

$\mathbf{b}' = (\leftarrow b'_i \rightarrow)^T$ A given vector of the right-hand-side of a mathematical program; also the fixed external-flows in a network-flow program

$\bar{\mathbf{b}}$ Updated right-hand-side of a linear program during a simplex procedure; also the birth rates in a cohort-survival analysis

$\mathbf{b}^1$ The portion of the right-hand-side corresponding to the complicated set of constraints in a mixed-integer-program

$\mathbf{b}^2$ The portion of the right-hand-side corresponding to the straightforward set of constraints in a mixed-integer-program

β A calibration constant, such as the positive exponent of a spatial cost-function or the round-trip factor in stochastic facility-location. (This same constant β is also referred to as *b*)

β' A calibration constant

β_i Current level of inventory at location *i*

β_t Prewhitened output time-series in a transfer-function model

B An arbitrarily large integer; also the backshift operator in a time series

B' Bifurcation set of control variables

B'' Blue-collar employment

B_k Percentage reflectance in band *k* of a satellite sensor

B_L, B_R Left and right boundaries of a firm's market area

B'_k Number of times a facility is exposed to demands in period *k*

B^k Bound value for distance from a vertex, used to locate the intersecting point q_k or a candidate location for a center

$B^M_{\text{Min}}, B^M_{\text{Max}}$	Lower and upper bounds for the border-line length of a subregion
$\mathbf{B} = [b_j]$	Birth matrix with nonzero diagonal-elements showing the "birth" rate within subarea j
$\mathbf{B} = [b_{ij}]$	Arbitrary matrix in a tableau of network-with-side-constraint program, corresponding to the flow variables
$\mathbf{B'} = [\beta_{ij}]$	Calibration-coefficient matrix in the first stage of a 2-stage least-squares, which measures $q \times k$, where q is the number of endogenous variables and k the exogenous variables
$\tilde{\mathbf{B}} = [\tilde{b}_{ij}]$	Quasi-deterministic transition-matrix in a compartmental model
$\mathbf{B}_i$	Diagonal-block i of the inverse of a network node-arc incidence-matrix, expressed in terms of a spanning subgraph
$\mathbf{B}'' = [b'_{ij}]$	Fixed cyclic-permutation δ' expressed in terms of a matrix operation, where $b'_{i,\delta'(t)} = 1$ and all other elements $b'_{ij} = 0$
$\bar{B}$	Initial basis for a network-with-side-constraint model
c	Cost of operation, unit-cost, or a constant in general (e.g., c_i is the unit cost at location i; c_{kl} is the "interaction cost" of moving materials between workstations k and l in an assembly line)
c'	Proportionality constant
c^k	Weight reflecting the relative importance of home-based retail-trips for purpose k
${}^r\mathbf{c}^s(\mathbf{x})$	rth-stop coverage of state s by routing-variable $\mathbf{x}$
$\mathbf{c} = (\leftarrow c_j \rightarrow)$	Cost vector in the objective function of a linear program, which is also the gradient of the objective function; here c_j is the constant unit-cost
$\mathbf{c}'$	Consumption-coefficient vector, whose ith element is the ratio of the purchased-value-of-the-commodity-from-the-ith-industrial-sector to the household income
$\mathbf{c}_B$	The part of the cost-vector $\mathbf{c}$ corresponding to the basic variables
$\mathbf{c}_N$	The part of the cost-vector $\mathbf{c}$ corresponding to the nonbasic variables
$\mathbf{c}^r$	Binary vector of rth-stage coverage-requirements in the decomposed recursive-program
$\mathbf{c}^{k+r}(k)$	Binary vector of rth-stage coverage-requirements on each origin—destination pair in cycle k; $\mathbf{C}(k) = [\leftarrow \mathbf{c}^{k+r}(k) \rightarrow]$
conv($\tilde{Q}'$)	Convex combination of discrete points $\tilde{Q}'$ in a feasible region of an integer program
C	Generalized cost to include both time and monetary outlay, or unit composite-cost in general (e.g., C_i is the generalized cost of operation or the inventory-carrying cost at location i, C_{ij} is the composite transportation-cost from location i to j, C^p_{ij} is the composite transportation-cost from location i to j for commodity p etc.)
C'	Number of columns in a lattice, grid or a pixel image; also household expenditure on community amenities (which is part of non-locational expenditure)
C_0	Overhead of a firm
C_o	Operating cost
C_s	Capital cost
C_j	Equity factor in districting algorithms

C_X	Coefficient-of-variation of variable X, or $s_{X/}\ \overline{X}$
C_{XY}	Cross-covariance between random variables X and Y
$C(C_{ij})$	Propensity, distribution, or accessibility function between i and j, assuming such forms as exponential function or power function of spatial-cost C_{ij}
$C[a](\mathbf{x})$	Performance of arc or path a as a function flow-vector $\mathbf{x}$
$C'(\tau)$	Accessibility to work-opportunities as a function of time τ
$C^k(\tau)$	Accessibility to the kth non-work-opportunity as a function of time τ
$C_i(\cdot)$	The cost function (including land rent), or performance function, of firm i—expressed in terms of the supply volume V_i^s or other arguments
$C_{ij}(V_{ij})$	Transportation cost between origin–destination pair $i - j$ as a function of flow V_{ij} between them
$C^{k,l}$	Transportation cost between origin k and destination l
$C^{mn}(r)$	Connectivity requirement between origin–destination pair m–n via at most rth-stop itineraries
$\mathbf{C} = [C_{\mathrm{ij}}]$	Arbitrary matrix in a tableau of network-with-side-constraint program, corresponding to the non-flow variables; also the covariance matrix
$\mathbf{C} = [\mathbf{c}^1, \ldots, \mathbf{c}^q]^T$	A $q \times n$ matrix of cost coefficients in a multicriteria linear-program, where each criterion j has a cost and a gradient vector $\mathbf{c}^j$
$\mathbf{C}(\cdot)$	State-connectivity function linking to past decisions and connectivity requirements in a recursive program
$\mathbf{C}'$	Diagonal matrix converting the gross-output vector to value-added vector
$\hat{\mathbf{C}}$	Matrix of estimated coefficients in stage 1 of 2-stage least-squares, measuring $q \times k$
$\overline{C}$	Number of cell columns in a grid region or in a raster image
γ	Unit price at the market, Lagrange multiplier, and a calibration constant in general
γ'	Capacity-utilization rate, bounded between zero and unity
γ_j^{pq}	Dual variable associated with the input–output coefficients in an entropy-maximization model
$\boldsymbol{\gamma}' = [q'_{\mathrm{j}}]$	Matrix of subareal growth-rates along its diagonal
$\overline{\gamma}$	Economic-base multiplier over a time-increment Δt, combining the activity-rate f and the population-serving-ratio a; $\overline{\gamma}_{ij}$ (with the subscript) would include the locational attributes as captured in work- and nonwork-accessibilities t_{ij} and u_{ij}
$\gamma_i(p, s)$	General 'strain' or the savings from including new-demand i via a triangular-inequality-style route-replacement between points p and s
Γ	The gross economic-multiplier deriving the total employment from the initial basic-employment
$\mathbf{\Gamma}$	Vector of economic-multipliers deriving the total employment in the study area from the initial basic-employment, including c_j, f and a
$\mathbf{\Gamma}_t$	Observation matrix in Kalman filter; when multiplied against the observed time-series, specifies what is actually observable
$\Gamma\ (W, p)$	Optimization results from a facility-location model where p facilities are relocated to respond to a maximum demand of W

$\mathbf{\Gamma}(k) = [\leftarrow \gamma_i(k) \rightarrow]$	Vector of payoff-function consisting of q entries, where $q \leq \acute{\boldsymbol{\eta}}\boldsymbol{\mu}$
d	Distance or spatial separation; also a proxy for a particular spatial order
d'	Amount of differencing to induce stationarity in a time-series
d''	A decision in a Markovian decision-process
d_i	Distance from location i (notice this is not necessarily Euclidean distance); or deviation from a standard or ideal in dimension i; also the capacity of arc i or the weights in a transfer function
d^k	Minimum threshold of retail-employment by trade-class k; d^R is the threshold for the case when there is only one trade class
$d_j(\mathbf{x})$	Multidimensional decision-boundary in a Bayesian classifier
$d(B) = d_0 + d_1 B$ $+ d_2 B^2 + \ldots$	Transfer function in a multivariate time-series, consisting of weights d_0, d_1, d_2, etc. and backshift operators B
d_{ij}	Euclidean distance or the spatial-cost in general between locations i and j
d_{ijk}	Euclidean distance or the spatial cost between locations i and j in state k
d_{ij}^h	Distance or travel time between nodes i and j by salesman or vehicle h
d^i	Time a salesman or vehicle visits node i in a tour or a route
d^{ij}	Distance or time between locations i and j, starting with arrival at i and terminating at arrival at j (notice this is not necessarily the Euclidean distance)
$d(\mathbf{i}, \mathbf{j})$	Planar Euclidean distance between two Cartesian coordinate points **i** and **j**
$d(\mathbf{x}_i, \mathbf{x}_{i+1})$	Spatial separation between consecutive stops $\mathbf{x}_i$, $\mathbf{x}_{\mathrm{i}+1}$
$\mathbf{d}$, $\mathbf{d}'$	Vector of arc capacities in network-flow programming
$\mathbf{d}^j$	Extreme direction along the jth axis in a linear program
$\mathbf{d}^k = (\leftarrow d_i^k \rightarrow)$	Direction of steepest ascent in the kth step of a hill-climbing optimization-algorithm, as characterized by n components of the vector
δ	Change in a quantity (e.g., δx is the increase or decrease in quantity x); δ_{ij} is the distance savings in directly going from i to j, instead of through an intermediate point k
$\delta(i)$	The steady-state decision whenever the state is i in a Markovian-decision-process
$\tilde{\boldsymbol{\delta}}$	Policy in a Markovian decision-process
$\tilde{\boldsymbol{\delta}}'$	Improved stationary-policy in the policy-iteration procedure of a Markovian-decision-process
δ^*	Optimal policy in a Markovian-decision-process
δ', δ''	Fixed cyclic-permutation
δ_i	Binary decision-variable to be switched on, conditional upon another decision-variable being engaged; also a calibration constant; or a nonnegative real-number denoting the number of legs in a subtour-breaking constraint
$\delta\Omega$	Boundary of the bounded-domain Ω
$\delta(k)$	Savings by using route k
$\delta^+(i)$	Set of nodes reachable from i
$\delta^-(i)$	Set of nodes incident upon i

δ_{ij} Route-distance savings by including demands i and j in a single, rather than separate tours, in accordance with the Clarke–Wright heuristic

$\boldsymbol{\delta}$ Vector of estimated-parameters in nonlinear regression

$\hat{\boldsymbol{\delta}}$ Least-squares estimate of $\boldsymbol{\delta}$, usually obtained as a conditional estimate

$\boldsymbol{\delta}_j = (\leftarrow\delta_{ij}\rightarrow)^T$ Orthonormal base of the transition-rate space when the system is in compartment j

D Distance or time of specified length

D' Data, population density, or a measure of crowding

D'' Dual polyhedron of a linear program; or a subset of nodes/vertices

D_{ab} Shortest distance from demand or customer a to demand b along a path, or along a tour from depot a to demand b

$D(i)$ Decision set in a Markovian decision-process

$D(a,b)$ Shortest distance along a vehicle route from terminal a to terminal b

D_i Cumulative distance (along a path) to demand i from a facility

$D_l(V_l^d)$ Demand at location l showing price against flow-quantity; in other words, price paid at demand quantity $V^d{}_l$

D'_i Cumulative distance (along a path) to demand i from all facility candidate-sites

D^k Total sales or service from facility k

D^H Upper-bound distance

D^L Lower-bound distance

D^H_j Maximum allowable household-density in zone j

$\mathbf{D} = [d_j]$ Death matrix with non-zero diagonal elements, showing the "death" rate within subarea j

$\mathbf{D'}$ Calibration-coefficient matrix in the first stage of 2-stage least-squares, measuring $q\times q$, where q is the number of endogenous variables

$\bar{\boldsymbol{D}} = [D_{ab}]$ $|I|\times|I|$ matrix of shortest cumulative-distances along a path from vertex a to vertex b

$\bar{\boldsymbol{D}}' = [D_{qk}]$ $|I|\times m$ matrix of distances from vertex q to arc k

Δ_i^j The difference between two utility measures i and j

$\nabla f(\mathbf{x})=(\leftarrow\partial f/\partial x_i\rightarrow)^T$
$=(G_x,G_y,G_z,\ldots)^T$ Gradient of a function over n variables

e The exponent value of 2.7183; also a calibration constant

e' Number of exogenous variables left in the econometric model after estimation

e'' Number of endogenous variables left in the econometric model after estimation

e_i Index to denote the ith type of industrial employment; also the ith arc in a network

e_{j_i} Arc j associated with node/vertex i

$\mathbf{e}^{i(j)}$ Unitary column-vector for arc j with unitary entry in the ith row

$\boldsymbol{\epsilon}$ A very small number or a random perturbation

$\boldsymbol{\epsilon}_k$ Efficiency-measurement error-term associated with the kth input–output pair in empirically curve-fitting a distance function

ε Normally-distributed error-vector with zero mean; when it has a constant variance, it could be a vector of random perturbations in the forecast using a transfer function, due to white noise in the inputs

E Total employment

$E^{'}$ Number of exogenous variables

E'' Number of endogenous variables

E^B Basic employment (E_j^B) is basic employment in zone j)

E^R Service employment

E^k Retail employment by trade-class k (E_j^k) is retail employment by trade class k in zone j)

$E(t)$ Relative smoothed-errors in adaptive-response-rate exponential-smoothing

$\tilde{E}_j$ Employment in the jth zone as projected from an areawide growth-rate for each sector

E_{ijk} Expected number of demands i in period k at location j

$E(i_1, i_2, h_1, h_2)$ Net change in travel-distance from an exchange of demands i_1 and i_2 between tours h_1 and h_2

$E^{'}(i_1, i_2, h_1, h_2)$ Modified generalized-savings-measure from an exchange of demands i_1 and i_2 between tours h_1 and h_2

$\mathbf{E}$ Row vector of employment-levels, made up of individual zonal employment E_i

f Average household-size in terms of the number of employed residents per household, or reciprocal of the labor-force participation-rate (also called the activity rate)

$f(\cdot)$ Regular function of the argument (e.g., the criterion function in dynamic programming)

$f(\mathbf{x}_q, \mathbf{x} - \mathbf{x}_q)$ A functional for which the directional derivative is being considered, approaching point $\mathbf{x}_q$ from point $\mathbf{x}$

$f^{'}$ Functional-attribute score, including spatial separation

$f^{'}(t)$ Cumulative demand at time-period t

f_i Demand-for-service frequency at location i; also the natural growth-rate of population in subarea i (the activity rate)

f^W Weighted activity-rate, where the weights are the percentages of regional population at each zone

f_{ik} Demand-for-service frequency at location i in state k

$f_{ik}^{'}$ Number of demands k serviced by facility i

f_i^t Convex combination of activity-rate f_i, where the weights are the normalized accessibilities into zone i

$f_j^{(l)}(\cdot)$ Speed-of-adjustment function for the jth zone and lth activity

f_r^{mn} rth-stop demand between origin–destination m–n

$\dot{f}(x) = df/dx$ Derivative of function f over variable x

$\mathbf{f}$ Partial-flow pattern in the decomposed RISE algorithm

F Set of candidate or new facilities to be sited, or an objective functional

$F(f(x)) = F(u')$ Fourier transform of function $f(x)$ in frequency $u^{'}$

$F^{'}(\mathbf{z})$ Production function with input rates $\mathbf{z} = (\leftarrow z \rightarrow)^T$

$F^{'}(\cdot)$ Regional-growth-rate function

F_k Fibonacci numbers; also the weighted activity-rate, with work-accessibilities from zone k as the weights

F_X	Derivative of function F with respect to variable X
$\dot{F} = \nabla F$	Gradient of the function F being maximized
F'_i	Unsatisfied demand or remaining service-capacity at each demand-node i to entertain additional vehicle-deliveries
F_{ij}	Accessibility factor between locations i and j, expressed as an inverse function of travel cost
F_{ik}	Probability that a demand from i is of type k
$\mathbf{F} = [F_{ij}]$	Square matrix of population-distribution rate by zone, measuring $n' \times n'$
$\mathbf{F}'(\mathbf{x}) = (\leftarrow F_i(\mathbf{x}) \rightarrow)$	A vector of functions whose interactions $\partial F'_i(\mathbf{x})_/\partial x_j \neq \partial F'_j(\mathbf{x})_/\partial x_i$ are asymmetric, where $\mathbf{x} = (\leftarrow x_i \rightarrow)^T$ for $i = 1, \ldots, n$
g	A scale factor; when serialized against argument m for example, $g(m)$, it is used to adjust zonal population from one loop of the Lowry model m to another $m + 1$, where $m = 1, 2, \ldots$
$g(\cdot)$	A special function of $\cdot$, such as the state equation; the relocation-cost function in stochastic facility-location; or the expected-master-travelling-salesman-tour length in probabilistic travelling-salesman-problem
g_k	Generalized unit-cost at facility k or for vehicle k
g'_i	Load to be picked up at node/vertex i
g''_i	Spatial "drift" of activities toward location i, in accordance with a profit/benefit motive or some gravitational potential-function
g_{ij}	Short-hand notation for nonwork accessibility between i and j
$\mathbf{g}$	Vector of coefficients associated with the discrete-variables $\mathbf{y}$; when used as a function, it is the subgradient
$\mathbf{g}(j) = (\leftarrow g_{h(j)} \rightarrow)^T$	Vector of input measures for a decision-making unit j
G	Number of salespersons in a travelling-salesman problem, or the number of vehicle-tours out of a depot
G'	Maximum fleet-size available at a depot; or share of the population which are immigrants
$G(\cdot)$	Multiple-travelling-salesmen expected-tour-length-function involving k salespersons
$G(\xi)$	Generating function for the probability distribution $P_0, P_1, P_2, \ldots, P_n$ where ξ takes on values of $0, 1, 2, \ldots, n$
$G(\boldsymbol{\xi}, t)$	Generating function for the probability distribution $P\,(\mathbf{X}^*_0, \mathbf{X}^*, t)$; where $\mathbf{X}^*_0$ is the initial-condition vector, $\mathbf{X}^* = [X^*_1(t), X^*_2(t) \ldots, X^*_n(t)]^T$, and where the n-dimensional-vector $\boldsymbol{\xi}$ takes on values of $\xi^{X^*} \equiv (\xi_1^{x_1^*}, \xi_2^{x_2^*}, \ldots, \xi_n^{x_n^*})^T$, for $\mid \xi_j \mid <1$. Thus for the stationary, irreducible Markov-process, it assumes the form $P(X^*_0) + \xi_1^{x_1^*}P[X^*_1] + \xi_2^{x_2^*}P[X^*_2] + \ldots + \xi_n^{x_n^*}P[X^*_n]$
G_i	Class or group i; also a generalized spatial-statistic for point i
$G_i(p,s)$	Generalized savings-measure from including demand node i between demand points p and s in a location–routing heuristic
$G'_i(p,s)$	Modified generalized-savings-measure from including node i between points p and s, after considering different depot-based tours
$G^*_i\,(h'')$	Net change in cost from displacing demand i from tour h to h''
$G^{**}_i(h'')$	Net change in cost from displacing demand i from tour h to h'' considering different fleets

G^{ij}	Transaction of goods and services between the *i*th and *j*th industrial sectors
G_{ij}	General location-pair spatial-statistic
G^{pq}_{ij}	Monetary transaction between the *q*th industrial sector in zone *j* and the *p*th economic-sector in zone *i* in an input-output model; with shorthand notation being $G^{pq}_{\cdot j}$ for consumption and $G^{q}_{ij\cdot}$ for production respectively, considering only the nonzero elements
$\mathbf{G} = [G_{ij}]$	The growth matrix showing the growth springing off from group/location *i* to group/location *j* (within a period of time); also a basic-feasible-solution to a simplex-on-a-graph
$\mathbf{G}(\cdot)$	Vector return-function in a recursive program
$\mathbf{G}' = [g_{hj}]$	Input matrix containing the *h*th input for decision-making-unit *j*
$\zeta^{(l)}_{j}(\cdot)$	Economic surplus- or deficit-function at zone *j* of the *l*th type
h	Index for a variable; generally to show a fleet type, a category of inputs (costs) in data-envelopment analysis, or the iteration number in a recursive program
h'	Minimum fleet size
$h'(\cdot)$	State-transition function in dynamic programming
h''	Calibration parameter in a dynamic version of a spatial-location model; an example is the time-scale parameter to convert activity to a rate-of-change
h_k	Height of a subregion *k*
h_{ij}	A rate- or calibration-constant in a deterministic compartmental-model; for example, the interaction between regions *i* and *j* in a multiple-region predictor–prey equation-set, or a shorthand notation for work-accessibility
$\mathbf{h}(j) = (\leftarrow h_{k(j)} \rightarrow)^T$	Vector of output-measures for target decision-making-unit *j*
H	Housing expenditure for a household (part of locational expenditure)
$H(\cdot)$	The Hamiltonian function in terms of the state equation, the costate or adjoint variable, and the figure-of-merit at the present; it also stands for a general function
H'	An upper limit of discrete index *h*
$H'(\cdot)$	Regional growth-rate function
H''	Set of vehicles in a fleet, or the set of vehicle types in the fleet
$\lvert H'' \rvert$	Cardinality of set *H*′, or the number of members in the set; here it is the fleet size
H^i	Transaction of goods and services to the *i*th household-sector
H_i	Set of potential tours in which demand *i* can be included
H_p	Cost of one dispatch on route *p*
H^G_r	Imports to region *r*
H'_i	Hazard a node *i* is exposed to
H_{ij}	Hazard a link (*i*, *j*) is exposed to
$H'_{ij}(\cdot)$	Flow-rate function from compartment *i* to compartment *j*
H^p_{ij}	Monetary transaction between the household sector in zone *i* and the *p*th economic-sector in zone *j* in an input–output model
η	Elasticity of demand
$\eta_{\alpha/2}$	100(1–α/2 percentile of the standard normal-distribution

θ	A parameter in general; for example, it can show decline in demand per unit-of-spatial-separation; θ_i is the rate-of-decline (or diffusion rate) of inflows into *i*
θ_t	Coefficient of the *t*th term in a moving-average time-series
$\theta(B)$	The backshift operation of a moving-average model
θ_{ij}	Proportion of activities (or trips) from origin-location *i* that end up in destination-location *j* based strictly on accessibility alone
$\mathit{\Theta}_{ij}$	A short-hand notation for the spatial-interaction term, indicating the proportion of activities (or trips) from origin-location *i* that end up in destination *j*—based on both accessibility and the attractiveness at the destination; i.e., the normalized accessibility-function between *i* and *j*
$\Theta_k = [\theta_{ijk}]$	A *k*th-order spatial-matrix of moving-average coefficients
$\Theta(B) = [\theta_{ij}(\mathrm{B})]$	A spatial matrix of moving-average operators
i, j	Indices for nodes/vertices; *i* normally stands for a demand node and *j* a facility node; or they can just be any counter
$i(k)$	Beginning node of arc *k*
$j(k)$	Terminating node of arc *k*
$\mathbf{i}$	Cartesian coordinates of a demand *i*
I	Set of nodes/vertices in a network
$I(d)$	The spatial-statistic Moran's-*I* for a particular spatial-order as defined by the distance-parameter *d*
$\lvert I \rvert$	Cardinality of set *I*, or the number of members in the set
I_k	Profit or income for facility *k*
I_N	Set of unlabelled nodes
I_D	Dual objective-function in recursive program
I'	Household or aggregate income
I'_t	Aggregate income at time *t*
I'_h	Set of potential demands for exchange, with an existing demand on the tour *h*
I''	Subset of potential demand nodes within the set *I*, where demands are non-zero
I_{p_k}	Any subset of nodes in the *k*th-stop route p_k
$I(i)$	Set of nodes/vertices which are input markets
$I(0)$	Set of nodes/vertices which are output markets
$I(t)$	0–1 indicator-sequence reflecting the absence and presence of an intervention, overlaying the transfer-function on top of the time-series
$I_{i\kappa}$	A binary variable assuming unity if the combination of facilities κ provides a satisfactory service to demand *i*
I_{Rx}	Total expected-mutual-information between the facility pattern in the region *R* and the demand spatial-pattern (when $x = I$), or between the facility pattern and an individual demand (when $x = i_k$); i.e., how probable the facility pattern is consistent with what is known about the demand pattern *I* or individual demand i_k
$I\,[\mathbf{X}(k), \mathbf{\Gamma}(k)]$	*k*th-stage payoff or objective-function of a recursive program, defined in terms of decision-variables **X** and constraint parameters $\mathbf{\Gamma}$
$I\,(\mathbf{P}; \mathbf{Q})$	Information that allows updating a prior probability-distribution **Q** to probability **P**

${}^{r}I^{s}(\cdot)$	Net-benefit function in a decomposed recursive-program
$\mathbf{I}$	Identity matrix
$j^{*}(k)$	Optimal facility location in state k
$\mathbf{j}$	Cartesian coordinates of a facility j
J	Subset of nodes/vertices in a network, generally the candidate sites for facility location
J_q	Set of candidate production sites
$\lvert J\rvert$	Cardinality of set J, or the number of members in the set
J'	A particular control-point in the bifurcation set
J''	The double values that the state variable assumes, corresponding to the control variable J' in the bifurcation set
$J(i)$	Set of Voronoi polygons adjacent to the ith polygon
$\mathbf{J}_k$	Basis k of a multicriteria linear-program
k	Index to show category k (e.g., Z^k is the k*th* activity); it marks a node, the commodity, the tree in a forest, or just serves as a counter
$k(\cdot)$	Equation for the control variable over time, expressed in terms of the state, the costate or adjoint variables
k_i	Calibration or scaling constant for zone i in a doubly-constrained gravity model; the Moran's-*I* or General Spatial statistic; alternatively, it is the propensity to save (invest)
$\mathbf{k}$	row vector consisting of 0, + 1, –1 entries marking an orthonormal base of the transition-rate space
K	A discrete or continuous constant, or the upper limit of running index k
$K(t)$	Capital-stock investment over time
K_i, K'_i	Trip-production and -attraction rate at zone i respectively
$\bar{K}_j^p$	A scaling constant; it ensures that the inter-sectorial and inter-zonal flows sum up to the non-labor input to the input-output table for sector-p and zone-j
$\dot{K}_r$	Instantaneous rate-of-capital-accumulation in region r
κ	Combination of three or more facilities that perform a certain function
κ'	The complement of the set κ
κ^h	Cost of operating vehicle h
κ_i^h	Marginal cost of serving demand-node i
K	Combination of three or more facilities
$l(T)$	Total cost of spanning-tree T, which is sum of the arc costs
l'	Discount rate (e.g., on the number of commuting trips, or traditionally in the time stream of cost or benefits)
l^i	Lower bound of a specified time window for a salesman or vehicle to visit node/vertex i
l_j	Calibration constant for zone j in a doubly-constrained gravity model
l_k	Spatial order of the kth autoregressive-term in a spatial time-series
$l^{h''}$	Ordered set of neighboring points (p, s) representing candidate tour h'
$l_{h''/i'}$	Ordered set of neighboring points (p, s) in tour h'' after removing demand i'
$l^{mn}(r)$	Length of an r-stop route originating in m and terminating in n

${}^{r}\mathbf{l}^{s}(\mathbf{x})$	Route-length vector at stage r and in state s of a decomposed recursive-program, expressed as a function of the decision variable $\mathbf{x}$
L	Nonempty subset of demand nodes/vertices, where a demand instance may be characterized by having actual demands realized in a node subset L of the network nodes/vertices I; the symbol also denotes twice the boundary length of a district
$\lvert L \rvert$	Cardinality of set L, or the number of members in the set
$\tilde{L}$	Length of the perimeter of a subarea
$\bar{L}$	The length of a queue, including the entity being served
$L(\cdot)$	Lagrangian or maximum-likelihood function
$L^{'}$	Probability that the location visited is the termination point for the trip
$L^{''}$	A calibration constant in a bivariate predictor–prey difference-equation-set
L_q	Queue length (excluding the one being served)
L_r	Regional labor-input-factor
$L^{(l)}x_i$	Spatial-lag operator on the value of spatial unit i, where l refers to the lth contiguity-class such as the lth-order neighbors; alternatively, we can write $\mathrm{L}^{(l)}x_i$ as a matrix operation to compute the weighted sum of the neighboring values of i contained in vector $\mathbf{x}$, or $(\mathbf{W}^{(l)})^T\mathbf{x}$. In general, $\mathrm{L}^{(l)}(\cdot)$ stands for spatial-lag operator of the lth-order, with the 0th-order operator reproducing the observation itself, or $\mathrm{L}^{(0)}(\cdot) = \cdot$
$\mathrm{L}_T(\cdot)$	Length of a master travelling-salesman-tour, constructed out of the set of nodes/vertices $\cdot$
L_{ij}	Error (in terms of a "loss measure") when a Bayesian classifier mis-assigns a multi-attribute observation $\mathbf{x} = (x_1, x_2, \ldots)^T$ to group j when it actually belongs to group i; usually $L_{ij} = 0$ if there is no error and $L_{ij} = 1$ if there is a misclassification
$L_j(\mathbf{x})$	Average misclassification error (in terms of a "loss measure") when assigning multi-attribute observation $\mathbf{x} = (x_1, x_2, \ldots)^T$ to group j; a couple of computational transformations of this measure are $L'_{\mathrm{j}}(\mathbf{x})$ and $L''_{\mathrm{j}}(\mathbf{x})$
$\mathbf{L} = (\mathbf{x}_L(q'_1), \mathbf{x}_L(q'_2), \ldots)^T$	Matrix containing the left eigenvectors $\mathbf{x}_L$
λ	Dual variable or Lagrange multiplier, with a specific (not necessarily feasible) solution $\bar{\lambda}$ and the optimal solution λ^*
λ'_i	A normalized weight, where $\Sigma_i \lambda'_i = 1$ unless noted otherwise
λ''	Arrival rate for a queuing process
$\lambda^k = (\leftarrow \lambda^k_i \rightarrow)^T$	The kth solution-vector in a Lagrange-relaxation procedure
λ'^*	Dual optimal-solution to the linear-program subproblem at the last iteration within Benders' decomposition
$\Lambda(\mathbf{J}_k)$	The weight cone for multicriteria linear-program, showing the $\lambda^{'}$-weight combinations that characterize a particular solution $\mathbf{J}_k$ among the nondominated set of solutions
m, n	Indices for dimension or for a node/vertex
$m^{'}$	A calibration constant in a bivariate predictor–prey difference-equation-set

m^* A critical bifurcation-value in a bivariate predictor–prey difference-equation-set

m^1 A collection of entities of characteristic 1; e.g., the number of complicated constraints in a Lagrangian-relaxation problem

m^j A collection of entities of characteristic j; e.g., the number of high-frequency direction finders in a bundle located at station j

m_k Spatial-order of the kth moving-average term in a spatial time-series

m_r Vehicle-fleet requirement at depot r, or the number of deployed vehicles at depot r

$\mathrm{m}'_i, \mathrm{m}''_i$ In- and out-movement rate to and from subarea i

$m(k)$ Median for a median-filter using a $k \times k$ mask

$m_1, m_2, \ldots, m_{k'}$ Groups of demand nodes to be served by route 1, 2, . . . , k', with $m_1 + m_2 + \cdots + m_{k'} \leq |I|$

$m'(q)$ Maximum shortest-distance from point q

m'_{ji} Binary variable that is "switched on" when demand i is allocated to facility j in a combined data-envelopment-analysis/location model; also the benefit valuation for such i–j pair

M Area specification for a districting model

M_i Maximum inventory carried at node i

M_{Max} Maximum number of nodes in a vehicle route

$\tilde{M}, \tilde{M}'$ A couple of matchings in a spanning-tree/perfect-matching heuristic for the travelling-salesman-problem

$M(t)$ Absolute smoothed-error (used in conjunction with relative smoothed-error) for adaptive-response-rate exponential-smoothing over time

$M(\Xi)$ Maximum of the weighted distances from the center candidates to each of the demands in the candidate facility-locations Ξ

M' Non-locational expenditure such as food, clothing, education, savings etc.

M'' A very large number or weight

M_{ij} Minor of a square matrix

$M(W, p)$ Simulation results of a facility-location model where p facilities are relocated to respond to a maximum load of W

$\mathbf{M} = [m_{ij}]$ Migration matrix showing the migration rate between locations i and j

μ Mean of a probability distribution

μ' Service rate of a queuing process; also the number of intermediate stops in the longest vehicle-route

μ_j Positive weights placed upon an extreme direction $\mathbf{d}^j$ in a linear program

$\mu_i, \boldsymbol{\mu}_i$ Mean of observations in group i in both scalar and vector form

$\mu^{(j)}$ Scaling constant of the error ϵ associated the value v being measured, resulting in $\nu^{(j)} + \mu^{(j)} \epsilon^{(J)}$

ν A collection of integer numbers

ν_i Route shape parameter (serialized by i) used in location-routing heuristics, assuming values such as 1 or 2

ν_t Noise series in a transfer-unction multivariate time-series

μ^p Dual variable associated with the control total of areawide-transportation-cost constraint in an entropy-maximization model

Ξ Collection of candidate facility-locations

$\Xi(X)$ Collection of all candidate facility-locations in the decision space X

$\Xi(\mathbf{y})$ Collection of candidate facility-locations which are open (i.e., for those locations where $y_j = 1$)

$\Xi(z)$ Collection of candidate facility-locations in the Z space, whose distance bounds are within z units

ξ As used in the Minkowski's distance function, it is the proportion by which factor inputs have to be reduced to reach the efficient point on the production frontier

n' The number of units in a spatial entity (e.g., the number of zones in a region, the number of subareas in a study area, or the total number of pixels in an image)

n_s Number of sides in a subareal polygon (e.g., in a Dirichlet tesselation)

$n(a, b)$ Number of stops between origin-terminal a and destination-terminal b

N Population or number of households (e.g., N_i is the population at location i)

N_j Number of pattern vectors from class G_j, or the number of nodes or pixel vectors belonging to class j

N'(large) A large number

$\bar{N}$ Total working population in the study area

N^p Population working in economic-sector p

N_j^c Capacity for residential development in zone j

N_i' Set of spatial units (including facilities) within a distance S from demand i

N_{ij} Binary decision-variables in a districting model, serving as a "pointer" across a district boundary separating a geographic sub-unit i and one that is not j; it is unitarilly value if subunit j is acquired and i is not

$\mathbf{N}$ Row vector of zonal population N_i

$\mathbf{N}(k)$ The nonbasic column associated with variable k in a linear-programming tableau

o_i Export share of region i

$O(l^k)$ Worst-case kth-polynomial computational-complexity for input-data-length l

O_i Export from the ith region

$\boldsymbol{O}'(\mathrm{P}) = \{\leftarrow O_i'(P)\rightarrow\}$ Orientation sequence of a path P, consisting of $+1$ and -1 entries, depending on the orientation of the arc in the path sequence

$\boldsymbol{O}^i$ Export from the ith industrial sector, measured in dollars

$\boldsymbol{O}_j^i$ Export from the ith industrial sector in subarea j, measured in dollars

$\mathbf{O} = (0\leftarrow O^i\rightarrow)^T$ Export vector in an aspatial input–output model, showing the convention that the first sector (the household sector) has no exports

$\mathbf{O} = (0\leftarrow O_j^i\rightarrow)^T$ Export vector in a spatial input-output model, where i is the economic sector and j is the subarea

p	An integer number for the number of facilities, the number of services provided, the index for the pth vehicle route, the parameter in the l_p-metric, or the autoregressive lag order in a time-series
p'	Number of facilities in a subset of the p facilities (i.e., $p' \leq p$)
p_f	Price of fuel
p_g	Price of the good
p_k	Price of a commodity k, with $\mathbf{p}$ standing for a vector of commodity prices
p_i'	Probability of adopting strategy i in a two-person game
$p^{(j)}(\cdot)$	Probability function of choosing alternative j, $j = 1, \ldots, n$
p_{ik}	Empirical probability that demand k patronizes facility i; or the probability of transitioning from state i to state k
$\hat{p}_{ik}$	Estimated value of p_{ik}
$p_{i\cdot}$	Empirical probability that a demand patronizes facility i
$p_{\cdot k}$	Empirical probability that a demand k is being served
$p_{j'}^{'q}$	qth factor-of-production input-prices at subarea j
p_k'	Number of facilities of the kth type (as used in a multi-product facility-location formulation)
$\overline{p}(t)$	Capacity expansion at time t
p''	Price of composite consumption good
p_{ijk}	*Conditional* probability that event-type i occurs at geographic-region j at time-of-day k
$\hat{p}_{ijk}$	Prediction of p_{ijk} based both on the hypothesized intervention model and historical data
$\breve{p}_{ijk}$	Analytical prediction of the *relative* probabilities $p_{ijk'}$, for field implementation as a transfer function
$\tilde{p}_{ijk}$	Relative probabilities after intervention probabilities have been implemented, using the transfer function $\breve{p}_{ijk}$
$\grave{p}_{ijk}$	Deseasonalized relative-probabilities after intervention probabilities have been implemented
$\mathbf{p} = (\leftarrow p^{(j)} \rightarrow)$	Perron vector whose components are positive and sum to unity
$\mathbf{p}_i = (\leftarrow \dot{p}_{ij}(t) \rightarrow)^T$	Vector of transitioning probabilities from state i to state j (where $j = 1, \ldots, n$)
$\dot{\mathbf{p}}_i = (\leftarrow \dot{p}_{ij}(t) \rightarrow)^T$	Time-derivative vector of probabilities transitioning transitioning from state i to state j (where $j = 1, \ldots, n$)
P	A path; also a set of vehicle routes generated for a network
P'	Potential surface for destination choice, whose derivative dP'/dC_{ij} is often operationalized by the trip-distribution function
P_D	Dual space of the linear-programming relaxation problem
$P(p)$	Probability that p servers are occupied (busy)
$P(\cdot)$	Probability of an event $\cdot$
P_i	Nearest location for demand or customer i; also the probability that the system is in state i
$P_i(t)$	Probability that the system is in state i at time t
P_k', $P_{(k)}$	Steady-state probability of being in state k
$P_{id''}$	Steady-state probability that decision d'' is reached while in state i
P_{ij}	Binary decision-variables in a districting model, serving as a "pointer" across a district boundary separating a geographic

	sub-unit i from one that is not j; it is unitarilly value if i is acquired and j is not
P_{ijk}	*Joint* probability of event-type i occurring in area j at time k, given that an event-type i occurred at time k
$\breve{P}_{ijk}$	Analytical predictions of p_{ijk} aggregated monthly, based on the hypothesized intervention-model
p^{mn}_{k}	Set of vehicle routes covering origin–destination pair m–n via k-stop itineraries
p^{mn}_{c}	Set of vehicle routes covering origin–destination pair m–n via connect itineraries
$\overline{P}$	Scale of a facility as represented by its capacity, capital outlay etc.
$\tilde{P}_l, \overline{P}_l$	Lower and upper bound of the supply at location l
$\overline{P}'$	Aggregate production-function with capital as input
$\mathbb{P}(\bullet)$	Logical predicate over the argument •
$P_j(p)$	Steady-state saturation-probability of all p service-units (in stochastic facility-location)
$\mathbb{P}_i = (\leftarrow P_i \rightarrow)$ or $(\leftarrow V_{ij} \rightarrow)$	Updated probability-distribution for each of the n' subareas or $\vert I\vert$ nodes written in a vector form; also can be the updated travel-vector between i and j, $V_{11}, V_{12}, \ldots, V_{ij}, \ldots, V_{\vert i\vert \vert j\vert}$, measuring $\vert I\vert \cdot \vert J\vert$ long
$\mathbf{P}(t) = (\leftarrow P_i(t) \rightarrow)$	Vector of the state probabilities $P_i(t)$; also the square matrix of transition probabilities over time
$\dot{\mathbf{P}} = (\leftarrow \dot{P}_i(t) \rightarrow)$	Time-derivative vector of state probabilities P_i (t)
$\mathbf{P}' = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$	Matrix containing independent eigenvectors $\mathbf{x}(q'_j)$, $j = 1, \ldots, n$
$\boldsymbol{P}_{t\text{-}1,\, t}$	Variance–covariance matrix for the difference between the observed and estimated Kalman-filter time-series-vector (or the estimation-error vector)
π_i	Dual variable in a network; such as the shadow price at node i, or a real number showing the amount of load carried on board a vehicle at node/vertex i
$\pi^{(j)}$	Probability that an individual reviews his/her choice of the jth compartment in a compartmental model
$\pi_{ij}(\cdot)$	Probability a given individual moves from compartment i to compartment j—as a function of, say, the state variable and time
π_i^j	Dual variable associated with the ith column of the spanning-tree ($j = 1$) or non-spanning-tree ($j = 2$) part of the basis (in a network-with-side-constraint tableau)
$\pi(\bullet)$	Permutation operator on the argument •
$\pi_{ij}(j \mid i, d'')$	The probability of transitioning from state i to state j during one period of the Markov process, given a decision d'' has been made
$\mathit{\Pi}^n$	n-dimensional transition-rate space
${}^{r}\boldsymbol{\pi}^{s}(\mathbf{x}, \mathbf{y})$	Vector gross-return-function of decisions $\mathbf{x}$ and $\mathbf{y}$ (in a decomposition implementation of recursive-program)
$\prod(\cdot)$	Vector of gross return-functions of decisions in a recursive program
$\prod_0(t) = (\leftarrow \pi_{i0}(t) \rightarrow)^T$	Vector of transition rates with the "outside world" over time

$\Pi = [\pi_{kl}]$	Transition-probability matrix in a Markov chain or compartmental model, with each entry denoting the given probability of transitioning from state k to state l; also the matrix of transition rates from state k to state l
$\tilde{\Pi}$	Matrix of transition rates from state k to state l, considering both arrival and service in a queue
q	Index to show a node number, center number, median number, number of substations, or the number of attributes, criteria, endogenous variables, eigenvalues, or differencing parameter in a time series
q_k	Candidate location for a center k
q_{ik}	Probability that an event-type i occurs at time k
q'	Eigenvalue, with q'_{Max} as the principal eigenvalue; also the growth rate of an area (with q'_j being the subareal growth-rate)
q'_i	Probability that strategy i is followed (in a two-person game); also the ith eigenvalue
$q_i(\cdot)$	Inventory-cost functions at demand-node i; or simply the unit cost-of-time (a constant) from demand-origin i
$\bar{q}_j$	Mean queuing delay
Q	Total economic-activity in the study area, such as consumption in dollars or number of trips executed
Q_i	Ratio of two accessibility definitions from location i
$\tilde{Q}_l, \bar{Q}_l$	Lower and upper bounds for the demand at location l
Q'	Total number of servers, or number of suppliers
$\tilde{Q}'$	Set of discrete points in the feasible region of an integer program
Q''	Cost per rejected demand in a loss-system location-model
$\bar{\mathbf{Q}} = [\bar{\gamma}_{ij}]$	A matrix of economic-base multiplier over a time-increment Δt
$\mathbf{Q} = (\leftarrow Q_i \rightarrow)$ or $[Q_{ij}]$	Prior-probability distribution for locating in each of the n' subareas (written in a vector form); or the vector of prior-travel between i and j, Q_{ij}
$\mathbf{Q}_{t-1}$	Variance–covariance matrix of the white-noise vector $\boldsymbol{\alpha}_t$
$\mathbf{Q}'$	The $\mathbf{X}^T\mathbf{X}$ data-matrix in the nonlinear regression of a STARMA model; where $\mathbf{X}$ is not explicitly given, and has to be numerically estimated
$\mathbf{Q}'' = [q_j]$	Matrix with eigenvalues $q_1, q_2, \ldots$ along its diagonal
r	Rent or mortgage, as part of locational expenditure (e.g., r^i is the rent for a unit of land i at a distance d_i from market, and $\mathbf{r}$ is the vector of rents among these land units)
r_0	Pearson correlation coefficient
r_k	Satisficing-level of criterion k; also the autocorrelation of lag-k in a time-series
r'^k	Land-consumption rate per retail-employee of trade-class k
r'	An l_p-metric deviational measure from a standard or an ideal
$r'(\mathbf{y}', \mathbf{x})$	Generalized-Leontief distance-measure, as a function of inputs $\mathbf{x}$ and outputs $\mathbf{y}'$
$r(\cdot)$	Spatial-separation or response-time function of argument $\cdot$; or the return function in dynamic programming
r'_0	Partial correlation coefficient
r'_k	Partial-correlation-coefficient of lag-k in a time series

r_j	The expected response-time of service unit *j*, consisting of mean queuing-delay and mean-travel-time to the demand
r_{ij}	Direct user-charge at facility *j* for user from origin *i*
$r\,(i, d'')$	Reward expected at state *i* by making decision d'' (in a Markovian-decision process)
r_{XY}, $r(X,Y)$	Sample (cross) correlation coefficient between random-variables *X* and *Y*
$r_{Y\mid X_iX_j\ldots}(X_k)$	Partial-correlation-coefficient between *Y* and X_k, given X_i, X_j, . . . are in the equation already
$r'_i(\cdot)$	Euclidean distance between demand *i* and a facility
$\hat{r}_{lm}(k)$	*k*th-order spatial-temporal-autocorrelation between the *l*th and *m*th neighbors of the subject site
R	A closed region in Euclidean 2-space; the set of *n* subregions $\{R_1, R_2, \ldots, R_n\}$; or the multiple correlation-coefficient
$R(J)$	The set of *n* subregions, each identified by its service-facility location $\mathbf{x}_i$: $\{R(\mathbf{x}_1), \ldots, R(\mathbf{x}_n)\}$
R_T	Total physical region made up of subregions $R_1, R_2, \ldots, R_n$; these regions can be of higher dimensions than the Euclidean 2-space
R^n_+	Domain of continuous non-negative variables in Euclidean *n*-space
$\lvert R(k^*)\rvert$	The area of the largest empty-circle with center at k^*, located at any vertex of the bounded Voronoi diagram
$\lvert R(\mathbf{x})\rvert$	The area of subregion $R(\mathbf{x})$; $\lvert R(\mathbf{x}^*_i)\rvert$ is the area of the optimal *i*th Voronoi polygon, with its facility at $\mathbf{x}^*_i$
R'	In stochastic facility-location models, R' is the required time in dispatching a special reserve-service-unit from a neighboring jurisdiction
R^2	Coefficient of multiple-determination in regression
$\bar{R}^2$	Coefficient-of-multiple-determination after adjusted for the degree-of-freedom
$R^2_{Y\mid X_1,X_2,\ldots,X_k}$	Coefficient of multiple-determination between *Y* and $X_1, X_2, \ldots, X_k$
$R'(\mathbf{y}')$	Set of input requirements $\mathbf{x}$ to produce $\mathbf{y}'$ in a production function
R''	The entire image or entire region
$R(+\mid-)$,$R(-\mid+)$	Finite predictor/prediction-space used in spatial-temporal canonical-analysis
R_i	Subregion *i* within the entire region R''; also the production in subregion *i*
R'_i	Normalizing constant in a spatial-interaction function, or the denominator of the function Θ_{ij}
R^p_i	Production output of the *p*th industry in zone *i*
$\bar{R}$	Number of row cells in a grid region, a raster image, or a lattice
R^i	Monetary output from the *i*th industrial-sector
R^i_j	Monetary output from the *i*th industrial-sector located in subarea *j*
$R_s(d)$	Norm deviate of the generalized spatial-statistic (analogous to the two-tailed *t*-statistic)
$\bar{R}^p_j$	The observed value of non-labor input to the input–output table for sector *p* and zone *j*
$\mathbf{R} = (yz^e \leftarrow R^i \rightarrow)$	Output vector in an aspatial input–output model, showing the production in each economic-sector, starting with output from

	the household (or labor) sector (measured in wages) and followed by the first, second, . . . industrial sectors i; here, the y and z^e symbols are defined at the back of this "List of Symbols"
$\mathbf{R} = (yz^e_j \leftarrow R^i_j \rightarrow)$	Output vector in a spatial input–output model, showing the subareal production in each economic-sector i, starting with the subareal output from the household (or labor) sector (measured in wages) and followed by the first, second, . . . industrial-sectors by subarea j; here, the y and z^e_j symbols are defined at the back of this "List of Symbols"
$\mathbf{R}' = [\mathbf{x}_R(q'_1),$ $\mathbf{x}_R(q'_2), \ldots]$	Matrix containing the right eigenvectors $\mathbf{x}_R$
$\mathbf{R}''$	Commodity-value-added output-vector
$\mathbf{R}_t$	Variance–covariance matrix of the measurement error (or noise) in a Kalman-filter time-series
ρ	Parameter or dual variable to account for the delivery-vehicle capacity
$\breve{\rho}(\tilde{\mathbf{B}})$	Spectral radius of matrix $\tilde{\mathbf{B}}$
$\rho' = \lambda''/\mu'$	Utilization rate of a server in a queuing system, or ratio of the arrival rate λ'' and service rate μ'
ρ''	Intensity of activity in a subarea
ρ_j	Utilization-rate of a service-unit j in stochastic facility-location; also the import rate of region j
ρ^p	Productivity-in-the-pth-economic-sector per unit-of-labor
ρ_{ij}	Trade coefficient between regions i and j
ρ^{pq}	Technical coefficients showing the transactions between the pth and qth economic-sectors in an input–output model
ρ^{pq}_j	Technical coefficient at the receiving-sector zone-j
ρ^{pq}_{ij}	Technical coefficients showing the transactions between the pth economic-sector in zone i and the qth economic-sector in zone j in an input–output model
$\boldsymbol{\rho}$	Matrix of technical or input–output coefficients $[\rho^{pq}]$, trade coefficients $[\rho_{ij}]$, or combined spatial-technical coefficients $[\rho^{pq}_{ij}]$
$\hat{\boldsymbol{\rho}}$	Diagonal matrix of trade coefficients, $[\rho_{ii}]$
$\boldsymbol{\rho}^j = [\rho^j_h]$	A matrix of economic-multipliers for the jth economic-sector, disaggregated by each zone-h
$\boldsymbol{\rho}_S, \boldsymbol{\rho}_T$	The consumption and production multi-sectorial components of the input/output-coefficient-matrix ρ, derived from row- and column-sum normalization of transaction flows respectively, with $\rho_S\ \rho_T = \rho$; the spatial, multi-subareal version assumes $G^{pq}_j/G^{\cdot q}_j = \rho^{pq}_j$ and $G^q_{ij}/G^q_{\cdot j} = \rho^q_{ij}$]
$\hat{\rho}_{XY}$	Population cross-correlation between random-variables X and Y
$\hat{\rho}^2$	Relative size of the variance; $(1 - \hat{\rho}^2)$ is the variance reduction
s	Source of a network
s_p	Autoregressive season-length in a seasonal time-series
s_q	Moving-average season-length in a seasonal time-series
$\underline{s}$	Prescribed frequency-of-visit at a node/vertex
s_X	Standard deviation of the random-variable X
$s(j)$	Sum of vertex(node)–arc(link) distances for facility j (the smallest sum identifies the general median)
$s'(j)$	Sum of point–arc distances for facility j (the smallest sum identifies the general absolute median)

s^2	Sample variance, with s being the standard deviation
s_{ij}	Length of the border separating geographic sub-unit i from sub-unit j; also the surviving ratio of cohort-group j from cohort-group i
s'	Average size of a site; or the ratio between the demand potentials at sites i and j
s''	Slack node/vertex in a network
S	A set of alternatives (e.g., the set of solutions that satisfies a predetermined goal or standard, the branch-and-bound search-space in a linear-programming relaxation etc.)
$S(\bullet)$	Sum-of-squares surface constructed out of the parameters • in nonlinear regression
S'	Consumers' surplus (or net benefit) to a tripmaker in making a trip; alternatively it refers to a predetermined maximum-service-distance in discrete facility-location
S''	Another set of alternatives (for example, the set after introducing a new alternative)
S_k	Set of demand vertices or nodes that would be covered by a center at q_k
$S_i(p', q')$	The increase (or savings) via a triangular-inequality-style inclusion (or exclusion) of demand i between the adjacent points (p', q')
$s_i(l_{h''}/i')$	Increase in travel distance from serving demand i via tour h'' (after the former-demand i' has been removed)
S^i	Marginal-cost function for path i
$S_l(V_l^S)$	Supply function showing price against flow quantity, in other words price charged at supply-quantity V_l^S; here, the supply quantity V_l^s is defined later in this List of Symbols
S'_{kj}	Unit benefit of assigning the kth activity (or activity from zone k) to zone j
S_{jk}	The kth site-specific attribute of the jth facility (such as the acreage of a state park)
$S^{k,l}$	Marginal-cost function between origin k and destination l
$\mathbf{S}_{ij}$	Vector of level-of-service variables between locations i and j, including such variables as travel time and travel cost
σ	Standard deviation of a probability distribution
σ^2	Variance of a probability distribution (see also the sample-variance s^2)
σ'	Vendor score or simply a constant in a model
σ_i	Real number showing the "odometer" reading of a vehicle at node/vertex i
$\sigma^2_{\hat{Y}}$	"Tilting" effect, as measured in terms of the variance, on the regression line (due to the randomness of the regression coefficients)
$\sigma^2_{M'}$	"Tilting" effect, as measured in terms of the variance, on the regression line—when an additional data-point x' is added to the regression
σ^2_Y	Total regression-based prediction- or estimation-error, as expressed in terms of the variance of the predicted- or estimated-values Y
$\sigma^2_{Y'}$	Total regression-based prediction-error, as expressed in terms of the variance of the predicted values Y'
$\sigma^2_{M^*}$	Variance of a normally-distributed set of residuals, around the sample regression-line at $X = x^*$

σ_{ij}^{pq}	Calibration coefficient such as the subareal investment-coefficient or the marginal capital-output-ratio, quantifying the multiplier effect of investment among economic sectors and between subareas
σ_j^2	Variance (or second moment) of service-time at service-unit j
$\sigma^h = (\leftarrow\sigma_i^h\rightarrow)$	Vector of dual-variables corresponding to the ith constraints defining the hth travelling-salesman-polytope
$\boldsymbol{\sigma} = [a_j]$	Zonal population-serving-ratios along the diagonal of an n′×n′ matrix
$\Sigma = [\mathrm{cov}(\mathcal{E}_i\mathcal{E}_j)]$	Error covariance-matrix
t	Time dimension or simply a counter for a series of data (e.g., $N(t)$ is the population at time t, Δt is a time increment)
t'	Subareal share of transportation-accessibility-to-employment
t_b	Student-t statistic for calibration-parameter b
$t_{\alpha/2,\,n-2}$	t-statistic at $100(1 - \alpha)\%$ confidence-level and $n - 2$ degrees-of-freedom
t_N	Sink node/vertex of a network
t''	Technical-attribute score
t^k	Step size in iteration-k of a hill-climbing optimization-algorithm
t_0	Dwell time at a terminal
t_j^h	Delivery- or dwell-time at node j by salesman or vehicle h
t_{ij}	Normalized work-accessibility-function between i and j
${}^rt^s(\mathbf{x}, \Phi, \mathbf{V})$	Cost of providing service at state s and stage r of a recursive-program
$\tilde{t}$	Random service-time on-scene $\tilde{t}_i$ or off-scene $\tilde{t}_j$
$\bar{t}$	Expected value of on-scene service-times to all demands i
$\bar{t}\,'$	Ratio between intra-nodal distances at i and j
$\bar{t}_j$	Average service-time for a service-unit stationed at depot j, consisting of on-scene service-time at the demand $t^1{}_i$ and the off-scene service-time at the depot t_j^2
$\mathbf{t} = [t_{ij}]$	Matrix of normalized work-accessibilities, measuring $n' \times n'$
$\mathbf{t}^k = [\tau_{ij}^k]$	Matrix of travel-times between i and j
τ	Time duration (e.g., τ_{ij} or $\tau(i, j)$ is the travel time from location i to j)
τ'	Calibration constant in a dynamicized input–output model
τ_k	A user-defined scalar in the subgradient optimization routine ranged (say) between 0 and 2
$\tilde{\tau}$	Random variable for service-time in a queuing process; $\tilde{\tau}_{j\mid i}$ is the random service-time for demand i from depot j
$\bar{\tau}_j$	Expected one-way travel-time to a random demand from depot j
$\bar{\tau}_j'(k)$	Expected travel-time from j to all demands in state k
T	Transportation cost as part of locational expenditure; also quantifies other technological factors
$T.$ or $T(\cdot)$	A-priori travelling-salesman-tour as a function of ·
T'	Minimum spanning-tree of a graph
T''	Multi-graph, derived from the minimum spanning-tree by duplicating every arc of the graph; also an instance of the travelling-salesman problem
T_N	Alternate sink-node/vertex in a network for excess flows
T_j	Number-of-neighbors surrounding geographic sub-unit j

T'_i	Proportion of sales from subject location to demand at *i*
T''_i	Electrical-flow capacity of a substation *i*
T_{ij}	Number of *i*th-group neighbors for a *j*th-group geographic sub-unit
$\hat{T}_{ij}$	Current estimate on random-variable T_{ij}
$\mathbf{T}$	Diagonal matrix of zonal activities such as population
$\mathbf{T}(\cdot)$	Vector of cost-functions in a recursive-program
$\mathbf{T}_B$	Basis for a simplex-on-a-graph, represented graphically as a tree
u	Accessibility-to-population, or a calibration parameter in general; for example, u_{ij} is the normalized nonwork-accessibility between *i* and *j*
$u(t)$	The set of infinite control-paths between the initial point $t = a$ and end point $t = b$
u'	Frequency of a signal
u''	Ratio of the maximum travel-distances between nodes *i* and *j*
$u_i(t)$	Dual variables in a recursive-program for $t = 1, 2, \ldots$
u^i	Upper bound of a specified time-window when a salesman or a vehicle visits node/vertex *i*
$\overline{u}_{ij}$	Capacity on arc (i, j) in a network
${}^{r}u^{s}(\mathbf{x}, \mathbf{y})$	Inference dual-variable to show the value (or shadow price) of relaxing an *r*th connectivity-requirement at state *s*
$\mathbf{u}$	Surplus variables in a linear program; also a subset of control-variables **U**
$\mathbf{u}' = [u_{ij}]$	Matrix of nonwork accessibilities, measuring $n' \times n'$
U	Utilities (e.g., U^* is the maximum amount of utility from a given income or budget)
$U(h)$	Route length or the range of a vehicle tour for vehicle type h $(h = 1, 2, \ldots)$
U'	Maximum route-length or range among a fleet of vehicles, $U' = \text{Max}_h[U(h)]$
$U(t)$	Control variables over time *t*
$\mathbf{U} = (\leftarrow U_j \rightarrow)$	Vector of control-variables in control theory (slow variables), usually expressed as a function of *t*; U_j also stands for just the *j*th canonical-variate
$\mathbf{U}$	Diagonal matrix of zonal activities such as employment
$\mathbf{U}(k) = [\leftarrow \mathbf{u}^{k+r} \rightarrow]$	Matrix of inference dual-variables in a binary recursive-program
v	Value or utility function, or simply the metric resulting from such a measurement
v_{ij}	The composite travel-cost, or the "utility function," between zones *i* and *j*, combining time, cost and other travel impedances into a single metric
$v(k)$	Average filter using the *k*th-order neighbors
v'	A given parameter (such as housing subsidy per household)
v''	Velocity of a service vehicle in stochastic facility-location
v_i	Dual variable associated with node/vertex *i*
v_w	Walking speed
v_{Max}	Maximum velocity of a vehicle
$v^{(j)}(\cdot)$ or $v^{j}(\cdot)$	Deterministic value-function for alternative *j*
$\overline{v}_{ij}$	The reduced cost for arc (i, j) in network-flow programming

$\mathbf{v}_i = (\leftarrow \nu_i^j \rightarrow)$	An eigenvector consisting of as many entries as the number of alternatives; this is equivalent to $\mathbf{x}_i = \mathbf{x}(q_i')$
$\mathbf{v}$	Surplus variables in a linear program
V	The amount of economic activities, traffic flow or patronage (e.g., V_i is the amount of activities or trips originating or terminating at location i, and V_{ij} is the exchange of economic activities or traffic movement between locations i and j); $\hat{V}$ is the estimated value and V^* is the observed value.
$V(h)$	Capacity of vehicle-type h, where $h = 1, 2, \ldots$
$V'(h)$	Capacity remaining on each vehicle h
$V'(\cdot)$	Normalized vehicle-capacity
V^d	Inverse demand-function, or the price schedule expressed as a function of a firm's (firms') total output; V_i^d is the excess demand at subarea i
V'	Set of vertices or nodes in a graph or network
V_i	The ith canonical-variate
V_{ij}	Flow between origin-destination pair i–j; $\tilde{V}_{ij}$ is the lower bound and $\bar{V}_{ij}$ is the upper bound
V_{ijk}	Probability that a demand i of type k is received by facility j
V_{ij}^k	Trips of type k from i to j
$\hat{V}_{ij}$	Predicted interactions between subareas i and j
$\tilde{V}_{iq}^d$	Amount supplied by all the firms other than q to demand-location i
V_i^s	Output of firm i; also standing for the excess supply of a firm located in subarea i
ϕ	Calibration constant representing such parameters as the trip-generation rate or response rate of the system
ϕ^h	Polytope (feasible region) defined by the hth travelling-salesman-problem
ϕ'	Probability distribution (e.g., probability that the surplus resulting from the trip to j has a value in the neighborhood of S')
Φ	Cumulative distribution (e.g., $\Phi(\nu) = [F(\nu)]^n$ is the cumulative distribution-function of the largest-utility v among n independent samples; $\Phi_{ij}(S')$ is the cumulative-distribution-function of the surplus accruing from the preferred (optimal) trip between location i and j)
ϕ_k	Coefficient of the kth-lag term in an autoregressive-time-series
$\phi(B)$	The backshift operation of an autoregressive model
$\hat{\phi}_k$	Partial-autocorrelation-coefficient for the kth-lag term in an autoregressive-time-series
$\hat{\phi}_{kl}$	Partial-autocorrelation-coefficient at temporal-lag k and spatial-lag l in an autoregressive spatial-time-series
$\boldsymbol{\phi}(\cdot)$	Flow-vector function at stage s of a decomposed recursive-program
$\boldsymbol{\phi}(\mathbf{x})$	Demand density-function on Voronoi polygons
$\boldsymbol{\phi}$	Vector of pertinent flows at stage r and state s of a decomposed recursive-program; these flows can be expressed in terms of the pertinent demand-vector $\mathbf{f}$
$\boldsymbol{\phi}^T = (\leftarrow \phi_i \rightarrow)$	Vector of autoregressive coefficients in a conditional spatial-econometric model

$\Phi^k = [\phi_{ijk}]$ A spatial autoregressive-coefficient matrix of order k

$\Phi(B) = [\phi_{ij}(B)]$ A spatial autoregressive-operator matrix

$\Phi^k(\cdot) = [\leftarrow \Phi^{k+r} \rightarrow]$ Matrix of flow-vectors $\Phi^{k+r}\,[\leftarrow \mathbf{x}^{k+r}(k) \rightarrow]$

$\Phi^{k+r}\,[\leftarrow \mathbf{x}^{k+r}(k) \rightarrow]$ Flow-vector at the kth cycle and rth stage, showing origin-destination-connectivity as a function of the iterative multi-stop routing-decisions

X^2 Chi-square statistic

$\boldsymbol{\varphi}$ Expected cost between stockout and storage in a newsboy problem

$\boldsymbol{\varphi} = [f_j]$ Zonal activity-rates along the diagonal of the $n' \times n'$ matrix

ψ Value of a given function; e.g., Sierpinski's-curve value

ψ_j Weights used in time-series forecasting

$\psi_{k=1}^{\,n}\ (l_k)$ Dynamic-program recursion-function for computing the shortest-route-length l

Ω Dual variable corresponding to the terminal capacity constraint—a parameter to account for the given warehouse capacity; also regular vector space

$\bar{\Omega}$ A bounded domain including the boundary $\delta\Omega$

$\Omega_q = \{\mathbf{x}_q\}$ A feasible region within the vector space Ω; e.g., a set of constraints in a spatial-equilibrium model, expressed in terms of the flow decision-variables $\mathbf{x}_q$ for each of the suppliers q

Ω_{ij} Percentage-change-of-patronage at facility j from the demands that originate at i

$\boldsymbol{\Omega}(B) = \Sigma_i \boldsymbol{\Omega}_i B^i$ Backshift operator containing the dynamic multipliers Ω_i in a set of dynamic simultaneous-equations

$\boldsymbol{\Omega}_{t-1,t}$ Transition matrix in a Kalman filter

$\acute{\eta}_k$ Connectivity requirement on the origin–destination pairs during the kth cycle

$\acute{\eta}_k(r)$ Connectivity requirement on a subset of the origin–destination pairs during the rth stage in the kth cycle; i.e., the number of constraint functions defining the local flow-pattern in a recursive program for the RISE algorithm

w A constant, or an aggregate weight-parameter, placed on a variable or an estimator-measure (such as Moran's-I, and its variance, plus the mean and expected variance of the general spatial-statistic)

w_k A constant or a weight placed on entity or attribute k; when these weights are normalized and summed to unity, we write $\Sigma_i \lambda_i = 1$

w^k Weight reflecting the relative importance of workplace-based retail-trips for purpose k

$\tilde{w}_1, \tilde{w}_2$ Weight parameters used in the formulas for the variance of Moran's-I

w'_k Width of a subregion k

w'_t A white-noise series, consisting of a sequence of uncorrelated random-variables, each with zero mean and finite variance; engineers consider them as independent "shocks" that are transformed by a "transfer function" to another time-series whose successive values are highly dependent.

w''_p Frequency on route p

w_{ij} Weight placed on the demand-facility pair i–j or the weight placed on arc flow (i, j), otherwise referred to as cost coefficients in the equivalent linear-program; also denotes the weight entry in a spatial-weight-matrix **W**, with $0 \leq w_{ij} \leq 1$

$w_{ij}(d)$ Binary valuations of w_{ij} when an activity at j is within a distance d from i

w_{ijp} Frequency on the (i, j) segment of route p

w_i^j Weight contribution toward criterion i by alternative j

${}^r\mathbf{w}^s(\phi,V)$ Vector of route-frequencies at stage r and state s of a decomposed recursive-program

$\mathbf{w} = (\leftarrow w_i \rightarrow)^T$ Eigenvector consisting of q entries—this is equivalent to $\mathbf{v}_i$ and $\mathbf{x}_i$; also the cost vector in a network-flow program

$\mathbf{w}^{(l)} = (\leftarrow w_{ij}^{(l)} \rightarrow)^T$ The vector of spatial-weights associated with the lth contiguity-class; an example is the weights associated with the lth-order neighbors—notice this is equivalent to the spatial operator $\mathrm{L}^{(l)}(\bullet)$

W White-collar employment; also work load or demand placed on a service-unit

W_i Size of demand or activity at i, which is proxy for development opportunity at the zone; $\mathbf{W}'$ is the vector of development-opportunities among all zones

W_q Delay time in queue

W_T Total time in system, including delay time in queue and the time being served

$\mathrm{W}(t)$ Rate of investment in new capacity over time

W'_i Revised size of demand or activity at i

W_{ij} Service-effectiveness weight expressed as a function of the separation between demand i and facility j; i.e., the further apart i and j are, the less effective it is for service to be rendered

$\bar{\mathrm{W}}_i^p$ Observed value of attractiveness or the opportunity of zone-i as a location for industry-p

W_i^h Observed zonal-residence attractiveness or opportunity

W_i^p Observed zonal-shopping attractiveness or opportunity

$\mathbf{W} = [w_{ij}]$ A $q \times q$ pairwise-comparison weight-matrix used in the analytic hierarchy process; also denotes the weight matrix in spatial econometric-models, measuring $n \times n$

$\mathbf{W}' = [w_{gh}]$ An $n' \times n'$ activity derivation-and-allocation matrix of Lowry–Garin model, with each entry denoting a zone pair $g - h$

$\mathbf{W}'' = [w_j]$ The diagonal matrix consisting of per-capita value-added productivity (wage rate)

$\mathbf{W}^j$ Activity derivation–allocation, transition or spatial-weight matrix for the jth activity in a Lowry–Garin model

$\mathbf{W}^{(l)} = [w_{ij}{}^{(l)}]$ Spatial weight-matrix for the lth-contiguity class; with the normalized spatial-weights sum to unity $\Sigma_j w_{ij}^{(l)} = 1$ and $\mathbf{W}^{(0)} = [\mathrm{w}_{ij}^{(0)}] =$ **I** or the 0*th*-order neighbors being the subject entry itself.

$\mathbf{W}_t$ Gain matrix in Kalman filter, representing the net percentage of measurement-error or noise that is left after filtering

$(\mathbf{w}^{(t)}\mathbf{y})_{-y_t}$ Preprocessing of data **y** by removing the subject ith-entry, and then replace it with a value resulting from “filtering” with a spatial-”mask” $\mathbf{W}^{(l)}$ of order l

x^* Sample observation or the optimal value of x

x'	A particular observation for the random-variable X
x'_t	Actual, accurate data in a Kalman-filter time-series (to be differentiated from what is observable)
$\mathrm{x}_0, \mathrm{x}'_0, \mathrm{x}''_0, \ldots$	Decision boundary between pattern groups 1 and 2, 2 and 3, 3 and 4, etc.
x_{ij}	Allocation of demand i to facility j; or flow from i to j
x^i	Flow on path i in a network
x^p_i	Equilibrium economic-activity at each subarea i and sector p
$\tilde{x}^p_i$	Projected sales of product p in subarea i
x_{ijk}	Allocation of demand i to facility j in state k
x^{mn}	Lost calls between origin–destination pair m–n
$x^{m,n}(C^{m,n})$	Demand-for-transportation between origin–destination pair m–n as a function of the transportation cost between them
x^{mn}_p	Binary link-allocation of demand between origin–destination pair m–n to non-stop route or itinerary p
x^{qp}_{ji}	Input of commodity-q from subarea-j in the production of commodity-p in subarea-i
x^{mn}_{mip}	Binary allocation of demand m–n on route p as indicated by the usage of segment (m, i) in the itinerary
$\mathbf{x} = (\leftarrow x_j \rightarrow)^T$	Vector of decision-variables, or empirical readings (such as change-in-accessibility for all the activities j)
$\mathbf{x}_q$	An interior point in the feasible-region Ω_q
$\mathbf{x}_t = (\mathrm{x}_1, \mathrm{x}_{2,} \ldots)^T$	Observed readings in a time-series
$\mathbf{x}'_t = (x'_1, x'_{2,} \ldots)^T$	Actual readings over time in a Kalman-filter time-series
$\mathbf{x}_L$	The left eigenvector of a square matrix
$\mathbf{x}_R$	The right eigenvector of a square matrix
$\mathbf{x}^i$	The ith discrete-point proposal in a branch-and-bound tree, corresponding to a constraint in the Lagrangian-dual linear-program
$\mathbf{x}^k$	The kth basic-solution in a linear-program, or the kth set of decision-variables (e.g., solution alternative in a branch-and-bound tree, the location of the kth-facility, or the routing decision-variables for the kth-vehicle)
$\mathbf{x}^0_s$	Coordinates for the origin of a trip
$\mathbf{x}^0_t$	Coordinates for the destination of a trip
$\mathbf{x}^{i_k}$	The ith discrete-point proposal in a branch-and-bound tree during the kth-step of the subgradient-optimization procedure of Lagrangian relaxation
$\mathbf{x}^*(\mathbf{x}^0_s)$	Nearest public-transportation terminal for a trip starting at origin $\mathbf{x}^0_s$
$\mathbf{x}^*(\mathbf{x}^0_t)$	Nearest public-transportation terminal for a trip terminating at destination $\mathbf{x}^0_t$
$\hat{\mathbf{x}}'_t = (\hat{x}_1, \hat{x}_2, \ldots)^T$	Estimated-values of the observations in a Kalman-filter time-series
$\mathbf{x}^{k+r}(k)$	The kth iterative multi-stop-routing decision-variables for the rth-stage
$\bar{\mathbf{x}}^{k+r}(k)$	Realized values for the kth iterative multi-stop routing-decision-variables in the rth-stage
X	The decision-variable X; or the decision or alternative space in multi-criteria decision-making
X'	The state space in Markovian decision processes

$\bar{X}$	Average of the independent random-variable X in a regression model
X^p	Control total of areawide transportation-cost for commodity p
$\mathrm{X}(t)$	Random variable for the state at time t
X_k	Random variable for the state at stage or time k
$X_i(\cdot)$	Accessibility from origin i to all destinations as a function of such parameter as travel cost
X'_{lj}	Activity-l's accessibility to zone j
X_{ij}	Observed patronage of facility j by demand from location i
X_i^k	Amount of activity k in zone i
$\mathbf{X} = \begin{bmatrix} \leftarrow \mathbf{x}_1 \rightarrow \\ \leftarrow \mathbf{x}_2 \rightarrow \\ \cdot \\ \cdot \end{bmatrix} = [X_{ij}]$	Exogenous- or independent-variable $n \times (k+1)$ matrix in ordinary-least-squares regression, corresponding to n observations and $(k+1)$ calibration-parameters
$\mathbf{X}(t) = (\leftarrow X_i(t) \rightarrow)^T$	Vector of state-variables in control theory (fast variables), expressed as a function of t in terms of the individual state variables $X_{\mathrm{i}}(t)$ for states $i = 1, \ldots, n$
$\mathbf{X}(0) = \mathbf{X}_0 = (\leftarrow X_i(0) \rightarrow)^T$	Initial condition of the state-vector at time 0 for states $i = 1, \ldots, n$
$\mathrm{X}^*_{\mathrm{Max}}(t)$	The most-likely state
$\boldsymbol{X}(k) = [\leftarrow \mathbf{x}^{k+r}(k) \rightarrow]$	Matrix of binary-decision-variables in a recursive-program during the kth-cycle and the rth-stage
$\mathbf{X}_l(\Delta) = (\leftarrow \mathbf{x}_{lj}(\Delta) \rightarrow)$	Activity-l's accessibility to individual-zone-j expressed as change in the regional-share-in-accessibility
$\mathbf{X}'' = (\leftarrow X''_{\mathrm{i}} \rightarrow)$	Stationary states in system of interacting differential-equations
$\mathbf{X}^j = [X^j_{gh}]$	A matrix of accessibilities between zones g and h for activity j
y	Wage rate for a household or total wages across the labor-force
y^*	Sample observation or the optimal-value of y
y'	Regression-based prediction corresponding to a given x'
y_p	The pth-component of the $\mathbf{y}'$-vector in a network-tableau
y_t	Ordinate of an observed-data-point in the series $t = 1, 2, \ldots$
$\hat{y}_t$	Estimated ordinate of an observed-data-point in the series $t = 1, 2, \ldots$
y_q	Household-wage expenditure on the qth industrial-sector
y_{jk}	Binary-decision-variable to assign facility to node-j in state-k
y_{ijk}	Binary-decision-variable to indicate that node/vertex-i is served by facility-j in state-k
y^{mn}_k	Binary indicator to show that there are k stops between origin–destination pair m–n
$\mathrm{y}_{u(k),\,\mathrm{v}(l)}$	Binary decision-variable to indicate moving a facility from node/vertex u to v as the state transitions from k to l
$\mathbf{Y} = (\leftarrow y_j \rightarrow)^T$	Vector of integer-variables in a mathematical-program, or simply a point within the regular vector-space

$\mathbf{y}_q$ A point other than $\mathbf{x}_q$ within the feasible-region Ω_q

$\mathbf{y}' = (\leftarrow y'_i \rightarrow)^T$ A vector of criterion-measures consisting of attributes i; also the updated or 'refreshed' column in a network-flow-tableau during the simplex-iterations

$\mathbf{y}''$ Interim solution in Benders' decomposition

$\mathbf{y}(k)$ The updated (or "refreshed") kth column in a network-tableau

$\mathbf{Y}^j = (\leftarrow y^j_i \rightarrow)^T$ A vector of criterion-measures for alternative j, or the jth group of y_i-variables (e.g., the delivery commitment of vehicle j toward demand i)

Y The decision-variable Y, or random-variable notation of the explanatory or dependent variable in ordinary-least-squares regression; also the regional income

$\bar{\mathrm{Y}}$ Mean of the random-variable Y

Y' Outcome or criterion space of multi-criteria decision-making; also the prediction random-variable in regression

Y'' The combinatorial space of the discrete-variables y_i

Y_{ij} A spatial-variable defined by the coordinates i and j—a variable that is related to its neighbors in both axes of this coordinate system; also the cross product showing the covariance between the observations at i and j

$\mathbf{Y} = (\leftarrow y_i \rightarrow)^T$ Explanatory- or dependent-variable vector in ordinary-least-squares regression, consisting of n observations; $\hat{\mathrm{Y}}$ denotes the estimated-values of random-variable Y

$\mathbf{Y}^{ij} = [\leftarrow y^{ij}_l \rightarrow]$ Binary parameters of each constraint-function in recursive programming (p' in total), where i is the state-index and j the stage-index; $\mathbf{Y}(k) = \begin{bmatrix} \uparrow \\ \mathrm{Y}^{s,k+r} \\ \downarrow \end{bmatrix}$

$\mathrm{Y}(\cdot)$ State-connectivity linkage-function of past decisions and available vehicle-capacity in a recursive-program

$\mathbf{Y}'$ Labor-force-value-added output-vector

z Objective-function of an optimization-problem; also used to denote the activity-generation rate

z' A bound on z

$z(j)$ Objective-function value of the jth alternative

z_c Largest demand-facility assignment-distance

z_i Amount of product or services sold at demand-point i; or a transformed observation from the raw-data Z_i

z_t Stationary time-series with zero mean

z_{IP} An integer-programming objective-function that is to be estimated by Lagrangian-relaxation

$\hat{z}_t$ Stationary time-series with non-zero mean; also the estimated-value in an adaptive time-series

z'_j Binary variable to denote the location of a facility at j; z_j is used after y_j when there is more than one type of facility to be located; also the optimal benefit of opening facility-j in a generalized p-median-problem (as defined in a subproblem of Lagrangian-relaxation solution

z^j_0 Amount-of-output produced at supply-facility or plant j

z^j_{0i} Amount-of-output produced at plant j and sent to output-market i

z_{ij}	"Trunk" traffic from supply-source i to distribution-center j
z_i^j	Amount of input-i used by plant-j
z^{e_i}	Employment by the e_ith-type industry
z_i^e	Number-of-households in zone-i employed by industry
$z_{ij}^{e_t}$	Supply-of-labor by household in zone-i to zone-j for employment by the e_ith-type industry
z_{L}^i	Lower-bound of objective-function-value at iteration-i
z_{U}^i	Upper-bound of objective-function-value at iteration-i
$z^{'}$	Lower or upper bound of objective-function-value
z_{ij}	Binary indicator-variable to show whether a multiattribute observation $\mathbf{x} = (x_1, x_2, \ldots)^T$ for a pixel of color j has been properly classified into group i; $z_{ij} = 1$ when it is properly classified into group i (or $i = j$) and $z_{ij} = 0$ when it is improperly classified ($i \neq j$). In vector notation for two groups i and j, we write $\boldsymbol{z}_i = (z_{ii}, z_{ij})^T = (1, 0)$; and the random variable corresponding to $\boldsymbol{z}_i = (z_{ii}, z_{ij})^T$ is $\tilde{\mathbf{z}}_i = (\tilde{z}_{ii}, \tilde{z}_{ij})^T$.
z'_{ij}	Impedance between zones i and j
z_{LD}	Objective-function-value of a Lagrangian-dual
z_{LP}	Objective-function-value for a linear-program relaxation
z_{LR}	Objective-function-value for a Lagrangian-relaxation problem
$\grave{z}_i$	Goods in storage at location-i
$\mathbf{z}$	vector of **Z** values induced for stationary and with mean set to zero; also stands for endogenous variables in an econometric model
$\mathbf{z}_j$	Vector-of-pixels **z** for group j in a Bayesian classifier
Z	Activity level (where the activity can be population, employment, gray values, or any economic or non-economic activity)
$Z(i)$	Expected-value of the decision made at state-i
$Z'(i)$	Expected-value of the improved-decision made at state-i according to Howard's policy-iteration
Z_j	Objective-function value or activity level at location-j
Z_t	Raw-data time-series before inducing stationarity
Z'_t	Actual, accurate data in a Kalman-filter time-series (to be differentiated from what is observable)
$\dot{Z}_t$ $\ddot{Z}_t$	First and second differencing of time-series Z_t
Z'	Preference structure in multi-criteria decision-making
Z''	Deviation-measures from the efficient-contour of unity in the Minkowski distance-function
Z_{ij}	Value of spatial-data at grid-point $i - j$; often simplified to read Z_j to stand for the spatial-data value at location-j
Z_j^l	Value of the jth spatial-data at spatial-lag l
Z_+^n	n-dimensional Euclidean-space of positive discrete-values
$\mathbf{Z} = (\leftarrow Z_i \rightarrow)$	Vector of exogenous-variables Z_i of such activities as population and employment in each zone-i; $\mathbf{Z}_0$ is the initial-values of **Z**
$\mathbf{Z}(t)$	Density or relative-frequency of the state-vector $\mathbf{X}(t)$; in other words, the normalized state-vector
$\mathbf{Z}_j = (\leftarrow Z_{ji} \rightarrow)$	The jth-activity assigned to zone-i
$\mathbf{Z}^i$	Vector of the *total*-population/employment activity-levels at time-period (iteration) i, with $\mathbf{Z}^0$ as the given final-period *basic-activities* (from which other activities are generated)

Solutions to Exercises and Problems

I. SOLUTIONS TO SELF-INSTRUCTIONAL MODULES

In the following pages, we will provide the solutions to selected exercises and problems. The first part documents the solutions to the following seven Self-Instructional Modules. As the reader may recall, each module was introduced as an exercise at the end of the respective chapters. The module itself is physically located on the CD/DVD, prepared in a format suitable for instructional homework assignments. In keeping with the modular concept of these assignments, the solution to each module is provided as a separate document.

Chapter 1	EMPIRICAL MODELING MODULE
Chapter 2	PROBABILITY MODULE
Chapter 3	PROBABILITY DISTRIBUTION AND QUEUING MODULE
Chapter 4	GRAPH OPTIMIZATION MODULE
Chapter 5	RISK ASSESSMENT MODULE
Chapter 6	LINEAR PROGRAMMING MODULE Part 1—Model Formulation
Chapter 7	LINEAR PROGRAMMING MODULE Part 2—Solution

A. Empirical Modeling Module: Answers to Illustrative Exercises

ILLUSTRATION (2)

The graph should be of the form:

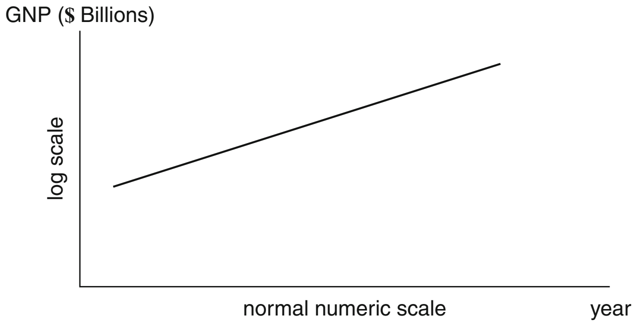


ILLUSTRATION (3)

Using *logGNP* = 0.064 t + 1.62, the following Table can be constructed:

year (t)	fitted *logGNP*
1	1.684
2	1.748
3	**1.812**
4	**1.876**
5	**1.940**
6	**2.004**
7	**2.068**
8	**2.132**
9	**2.196**
10	**2.260**
15	**2.580**
20	**2.900**
25	**3.220**
30	**3.540**

The fitted *logGNP* is converted to actual GNP and compared with the actual GNP:

year	actual GNP ($ Billions)	fitted GNP ($ Billions)
1	**53**	**48.3**
2	**59**	**56.0**
3	**68**	**64.9**
4	**68**	**75.2**
5	**85**	**87.1**
6	**97**	**100.9**
7	**116**	**116.9**
8	**142**	**135.5**
9	**166**	**157.0**
10	**197**	**182.0**

ILLUSTRATION (4)

year	0	5.2	10.4	15	20.8	26	31.2
cobalt 60 (*g*)	10	5	**2.5**	**1.25**	**0.625**	**0.3125**	**0.1563**

The graphs for cobalt 60 and cesium 134 should be of the form:

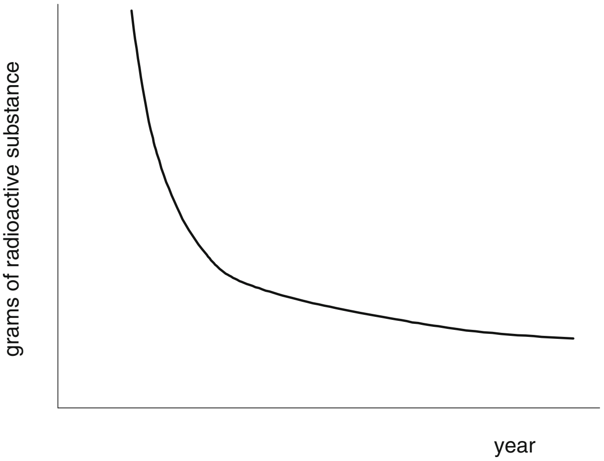
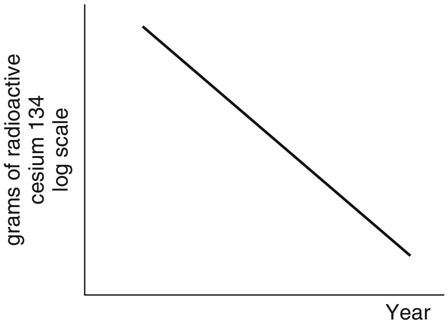


ILLUSTRATION (5)

The graph of cesium 134 on the semi-log graph should be of the form:



n	68.68	**54.48**	**45.89**	**40.03**	**35.72**	**32.40**
r (in %)	1	2	3	4	5	6

B. *Probability Module: Answers to Illustrative Exercises*

ILLUSTRATION (3)

S = {(H,H,H), (H,H,T), **(H,T,H)**, **(T,H,H)**, **(T,T,H)**, (T,H,T), **(H,T,T)**, (T,T,T)}

ILLUSTRATION (4)

S = {(M,D), (M,R), **(M,O)**, **(F,D)**, **(F,R)**, **(F,O)**}

ILLUSTRATION (5)

The sample space is:

(1, 1)	(2, 1)	(3, 1)	(4, 1)	(5, 1)	(6, 1)
(1, 1)	(2, 2)	(3, 2)	(4, 2)	(5, 2)	(6, 2)
(1, 3)	(2, 3)	(3, 1)	(4, 3)	(5, 3)	(6, 3)
(1, 4)	(2, 4)	(3, 4)	(4, 4)	(5, 4)	(6, 4)
(1, 5)	(2, 5)	(3, 5)	(4, 5)	(5, 5)	(6, 5)
(1, 6)	(2, 6)	(3, 6)	(4, 6)	(5, 6)	(6, 6)

ILLUSTRATION (6)

"At least one head" {(H,H), (**H,T**), (**T,H**)}
"At least one head or one tail" {(H,H), (**H,T**), (**T,H**), (**T,T**)}
"two heads" {(**H,H**)}
"one tail" {(**H,T**), (**T,H**)}
"first coin is head" is {(H,H), (**H,T**)}
"second coin is tail" is {(H,T), (**T,T**)}.

ILLUSTRATION (7)

The event, "the sum of the spots on the 2 dice is 7," is {(6,1), (5,2), (**2,5**), (**4,3**), (**3,4**), (1,6)}.
The event, "number on the second die is twice the number on the first die," is {(1,2), (2,4), (**3,6**)}.
The event, "the number on the second die is larger than the number on the first die," is {(1,2), (1,3), (**1,4**), (**1,5**), (**1,6**), (2,3), (**2,4**), (**2,5**), (**2,6**), (**3,4**), (**3,5**), (**3,6**), (**4,5**), (**4,6**), (5,6)).
The event, "the number on the first die is 2," is {(2,1), (**2,2**), (**2,3**), (**2,4**), (**2,5**), (**2,6**)}.

ILLUSTRATION (8)

The event, "first coin shows head or at least one coin shows head," is {(H,H), (**H,T**), (**T,H**)}.
The event, "first coin shows head and at least one coin shows head," is {(H,H), (**H,T**)}.
The event, "at least one coins shows tail or at least one coin shows head," is {(**H,H**), (**H,T**), (**T,H**), (**T,T**)}.
The event, "at least one coins shows tail and at least one coin shows head," is {(**H,T**), (**T,H**)}.

ILLUSTRATION (11)

3rd selection has **28** possible choices.
4th selection has **27** possible choices.
5th selection has **26** possible choices.
The number of different course loads is: 30 × 29 × **28** × **27** × **26** = **17,100,720.**

ILLUSTRATION (12)

2nd toss has **6** possible outcomes.
3rd toss has **6** possible outcomes.
The number of different outcomes is: 6 × **6** × **6** = 216.

ILLUSTRATION (13)

2nd toss has **36** possible outcomes.
3rd toss has **36** possible outcomes.
The number of different possible outcomes is: 36 × **36** × **36** = 46,656.

ILLUSTRATION (16)

$${}_8P_4 = 8!/(\mathbf{8-4})! = \mathbf{1{,}680}$$

ILLUSTRATION (17)

The 9 permutations are

aa	b**b**	cc
ab	**ba**	**ca**
ac	**bc**	**cb**

"APPLICATIONS" SECTION

There are **365** possible days for the 3rd person's birthday.
There are **365** possible days for the 4th person's birthday.
There are **365** possible days for the 5th person's birthday.
${}_{365}P^5 = 365^5 = 6.48 \times 10^{12}$

There are **362** days for the 4th person's birthday.
There are **361** days for the 5th person's birthday.
${}_{365}P_5 = \mathbf{365}!\ /\ (365 - 5)! = 365 \times 364 \times 363 \times \mathbf{362} \times \mathbf{361} = 6.30 \times 10^{12}$

(b) $1 - {}_{365}P_{20}/{}_{365}P^{20} = 1 - 0.589 = 0.411.$

(c) $1 - {}_{365}P_{25}/{}_{365}P^{25} = 1 - 0.431 = 0.569$

C. Probability Distribution & Queuing Module: Answers to Illustrative Exercises

ILLUSTRATION (2)

Random variable X = number of dots showing	Associated prob. of random variable X
1	1/6
2	1/6
3	1/6
4	1/6
5	1/6
6	1/6

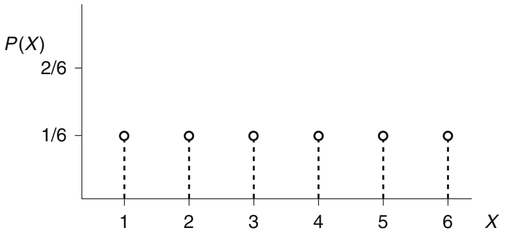


ILLUSTRATION (3)

X	$P(X)$	
0	**1/8**	(TTT) Zero head
1	**3/8**	(HTT), (THT), (TTH) One head
2	**3/8**	(HHT), (HTH), (THH) Two heads
3	**1/8**	(HHH) Three heads

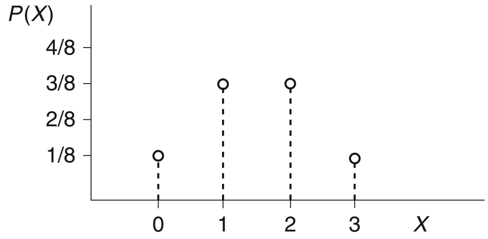


ILLUSTRATION (4)

(4-c)

$$n = 5, X = \mathbf{2}, p = \tfrac{1}{2}$$

$$P(X = \mathbf{2}) = \binom{5}{\mathbf{2}} (1/2)^2(1/2)^3 \ \mathbf{= 5/16 = 0.3125}$$

(4-d)

$$\mathbf{P(X = 5)} = \binom{\mathbf{5}}{2} \mathbf{(1/2)^5(1/2)^0 \ = 1/32 = 0.0313}$$

ILLUSTRATION (5)

(5-a)

$$n = \mathbf{10}, X = \mathbf{8}, p = 0.9$$

$$P(X = \mathbf{8}) = \binom{\mathbf{10}}{\mathbf{8}} (0.9)^8(\mathbf{0.1})^2 = \mathbf{0.194}$$

(5-b)

$$n = 10, X = 9, p = 0.9$$

$$\mathrm{P}(X = \mathbf{9}) = \binom{10}{9} (0.9)^9 \mathbf{(0.1)}^1 = \mathbf{0.387}$$

ILLUSTRATION (7)

(7-a)

$$m = 1, X = \mathbf{2}$$

$$\mathrm{P}(X = \mathbf{2}) = (e^{-1})\,(1^2)/\mathbf{(2!)} = \mathbf{0.184}$$

(7-b)

$$\mathbf{P(X = 3) = (e^{-1})\,(1^3)/(3!) = 0.061}$$

(7-c)

$$\mathbf{P(X = 4) = (e^{-1})\,(1^4)/(4!) = 0.015}$$

ILLUSTRATION (8)

(8-b)

$$P(X \leqslant \mathbf{2}) = 1 - e^{-0.4} = 0.330$$

(8-c)

$$P(X \leqslant \mathbf{3}) = 1 - e^{-0.6} = 0.451$$

ILLUSTRATION (9)

(9-b)

$$P(X \leqslant \mathbf{5}) = 1 - e^{-2.5} = 0.918$$

(9-c)

$$P(X \leqslant \mathbf{10}) = 1 - e^{-5} = 0.993$$

D. Graph Theory Module: Answers to Illustrative Exercises

ILLUSTRATION (1)

Vertex	A	B	C	D	E	F
Degree?	2	6	4	**4**	**2**	**6**

ILLUSTRATION (2)

path
cycle
3
3

A forest is graph without cycles. It is made up of trees, which are not necessarily connected.

ILLUSTRATION (3)

Vertex	A	B	C	D	E	F	G	H	I	J	K	L
Degree	4	4	4	**4**	**4**	**4**	**1**	**1**	**1**	**1**	**1**	**1**

It is a connected graph. It is not a tree. We can make it into a tree by eliminating one of the two arcs connecting A to F, B to C *and* D to E and eliminating arc AB.

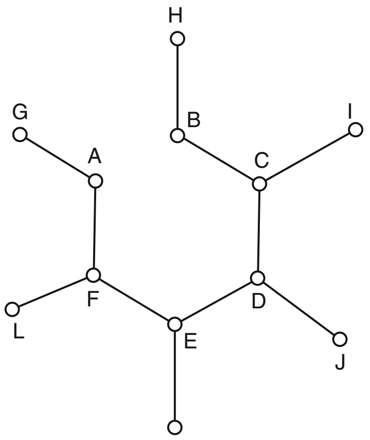


ILLUSTRATION (4)

A is 2 B is 2

ILLUSTRATION (5)

Yes, closed path can be traced. The degree of A is 2, B is 4, C is 2.

ILLUSTRATION (6)

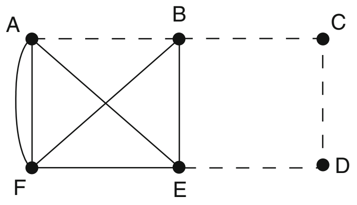
A is 2 B is 6 C is 2 D is 2

ILLUSTRATION (7)

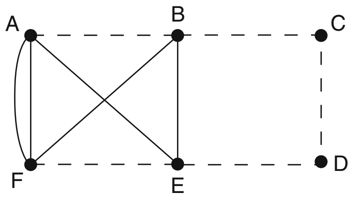
Vertex	A	**B**	**C**	**D**	**E**
Degree	2	**3**	**2**	**3**	**2**

The graph is semi Eulerian because *two* vertices, B and D, have odd number degree.

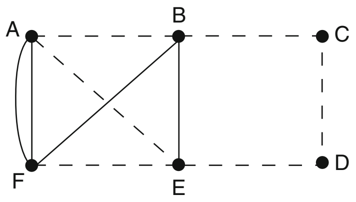
ILLUSTRATION (8)



A and E ⇒ Odd; B and F ⇒ Even

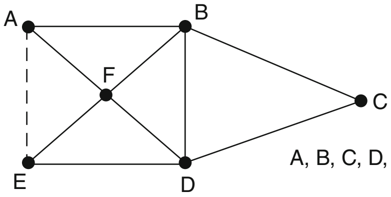


A and F ⇒ Odd; B and E ⇒ Even

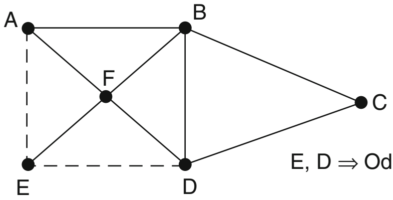


E and F ⇒ Odd; B and A ⇒ Even

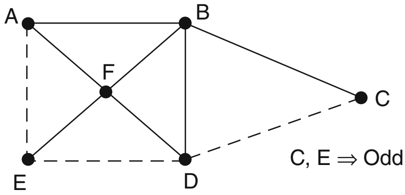
ILLUSTRATION (9)



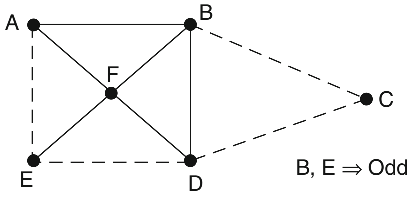
A, B, C, D, E, F ⇒ Even



E, D ⇒ Odd; A, B, C, F ⇒ Even



C, E ⇒ Odd; A, B, D, F ⇒ Even



B, E ⇒ Odd; A, F, D ⇒ Even

ILLUSTRATION (11)

The minimum spanning tree is

Arc	LC	CK	JE	BH	FG	CG	KB	AB	KJ	DE	IE
Length	1	1	1	1	2	**2**	**3**	**3**	**4**	**4**	**4**

Notice arc LF is skipped.

Total Length = 26 km

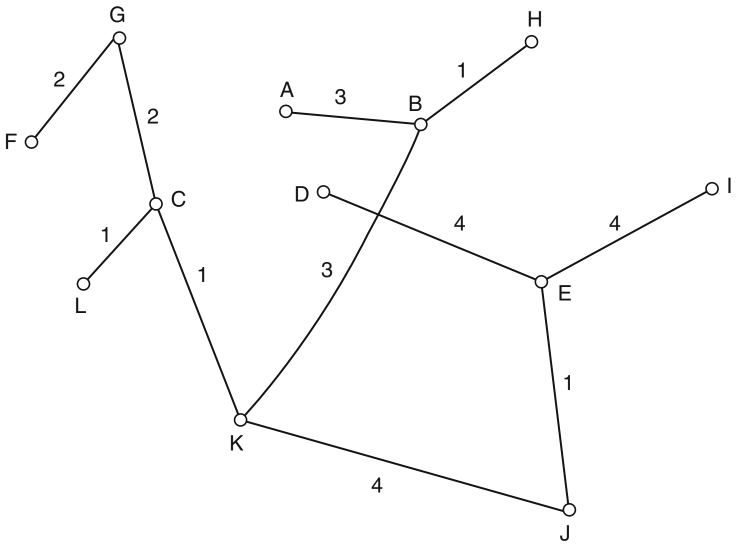
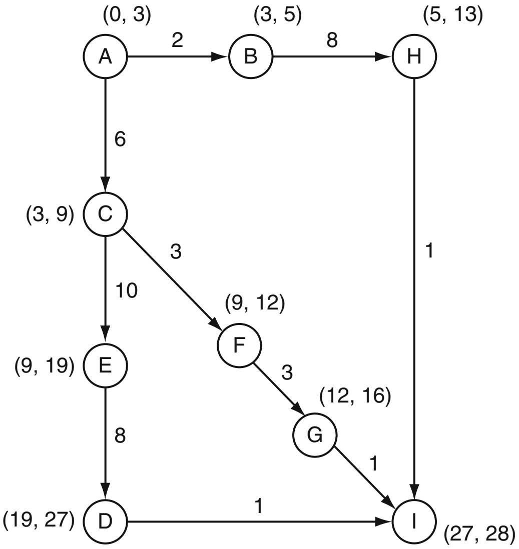


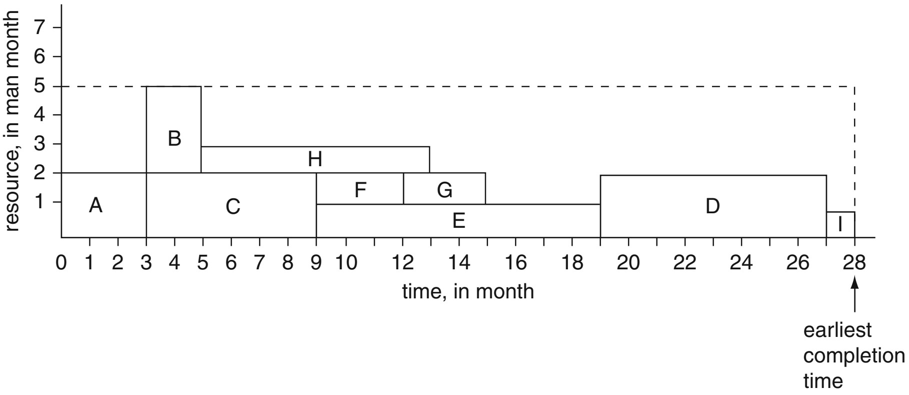
ILLUSTRATION (12)

Activity	Cost
C	$ 3,000
D	$ 4,000
E	$ 4,000
F	$ 3,000

ILLUSTRATION (13)



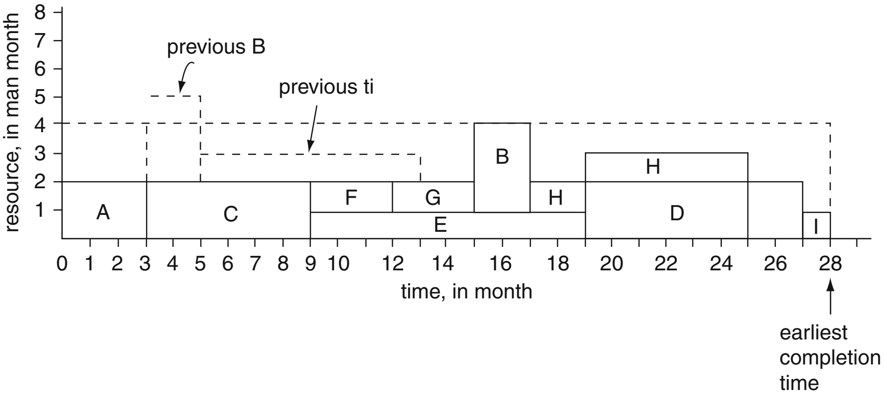
The critical path is A to **C** to **E** to **D** to I.



The intrinsic labor cost = 65 person $ 2,000 = $ 130,000.
The actual labor cost = 140 person months × $ 2,000 = $ 280,000.

Slacks:
Earliest start time for F is 9.
Latest start time for F is 21. Slack for F = 12
Earliest start time for H is 5.
Latest start time for H is 19. Slack for H = 14 months.

Resource Leveling



Total person months = 4 × 28 = 112 person-months.
Min actual project cost = (112 person-months) × ($ 2,000/person-month) = $224,000.
The resource leveling results in a saving of $ 56,000.

E. Risk Assessment Module: Answers To Illustrstive Exercises

On the benefit end, we have the net savings and the positive value of the project. On the cost end, we have all the expenditures. The net benefit is the difference between benefit and cost. Since the current project yields the greatest net benefit, it is to be built.

EXERCISES

Case 1:

$(750,000+150,000-500,000) = $400,000

Build, since the net benefit is a positive number.

Case 2:

Plant	Total benefit	Net benefit
A	**900**	**400**
B	750	**50**
C	**475**	**275**
D	**350**	**150**
E	**600**	**525**

Plant E has the largest net benefit.

Case 3:

Plant	Net benefit	Net benefit divided by initial cost	Cumulative initial cost, all plants
E	525	7.0	75
C	275	1.375	275
A	**400**	**0.8**	**775**
D	**150**	**0.75**	**975**
B	**50**	0.071	1,675

We propose to build plants E, **C**, **A**, **D**,

Case 4:

	Fatalities	Frequency
Nuclear plants	100	1/10,000
Air crashes, persons on ground	100	1/100
Chlorine releases	100	**about 1/75**
Fires	100	**1/10**
Dam failures	100	**about 1/50**
Air crashes, total	100	**about 1/7**
Total man caused	100	**about $^1/_2$**

ILLUSTRATION (2)

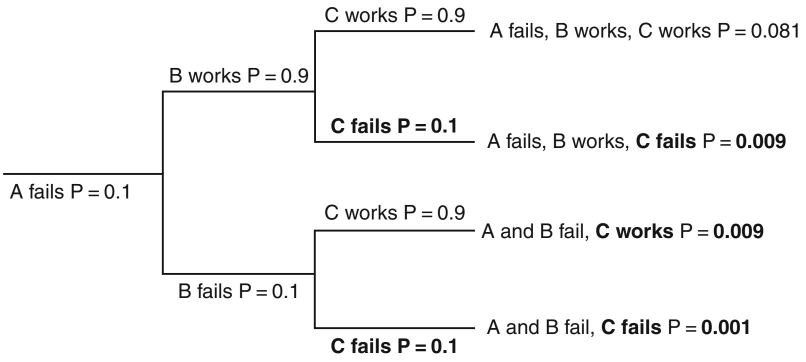
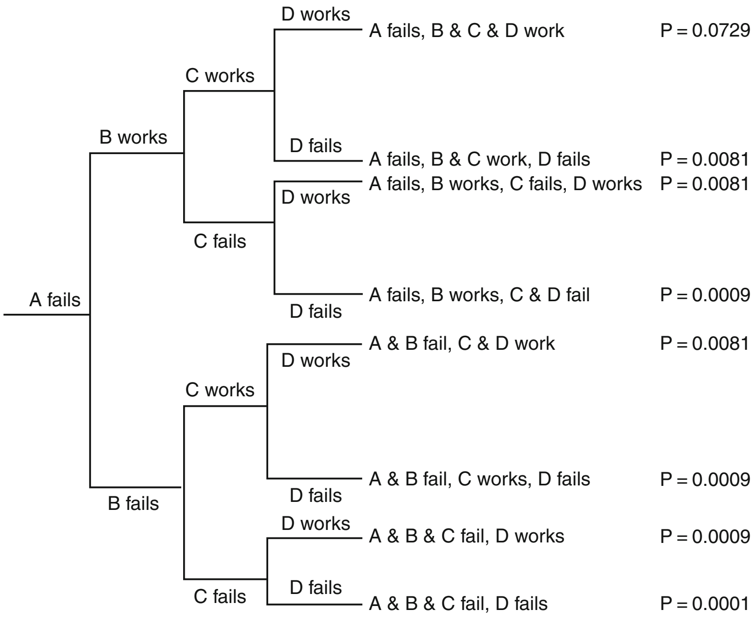

A fails P = 0.1
B works P = 0.9
C works P = 0.9
A fails, B works, C works P = 0.081
C fails P = 0.1
A fails, B works, **C fails** P = **0.009**
B fails P = 0.1
C works P = 0.9
A and B fail, **C works** P = **0.009**
C fails P = 0.1
A and B fail, **C fails** P = **0.001**

ILLUSTRATION (3)



F. Linear Programming Module: Part 1 - Modeling Answers To Illustrative Exercises

ILLUSTRATION (2)

$$\text{Max } Z = 20\ X_1 + \mathbf{14X_2} + \mathbf{10X_3}$$

constraint equations:

$$\begin{aligned} 6X_1 + 5X_2 + 3X_3 &\le 5{,}000 \\ 3X_1 + 3X_2 + X_3 &\le 3{,}000 \\ \mathbf{1}X_1 + \mathbf{2X_2} + \mathbf{3X_3} &\le 2{,}000 \end{aligned}$$

non-negativity:

$$X_1 \ge 0,\ X_2 \ge 0,\ X_3 \ge 0.$$

ILLUSTRATION (3)

$$\begin{aligned} \text{Min } Z &= 1X_1 + \mathbf{2X_2} + \mathbf{3X_3} \\ \text{s. t. } & 4X_1 + 3X_2 + 2X_3 \ge 6 \\ & 2X_1 + \mathbf{8X_2} + \mathbf{10X_3} \ge \mathbf{8} \end{aligned}$$

ILLUSTRATION (4)

X_{22} = **the number of units shipped from factory 2 to warehouse 2**

$$\begin{aligned} \text{Min } Z &= 20\ X_{11} + 25X_{22} + \mathbf{30X_{21}} + \mathbf{16X_{22}} \\ \text{s. t. } & X_{11} + X_{12} = 800 \\ & X_{21} + \mathbf{X_{22} = 600} \\ & X_{11} + X_{21} \le 750 \\ & X_{12} + X_{22} \le 650 \end{aligned}$$

$$X_{11} \ge 0,\ X_{12} \ge 0,\ X_{21} \ge 0,\ X_{22} \ge 0$$

ILLUSTRATION (5)

$$\begin{aligned}
\text{Min } Z &= 13.75\, X_1 + 11X_2 + 6.875\, X_3 \\
\text{s. t.} \quad & 220X_1 + 195X_2 + 110X_3 \leq 5{,}500 \\
& X_1 + X_2 + X_3 \leq 40 \\
& X_1 + X_2 + X_3 \leq 45
\end{aligned}$$

$$X_1 \geq 0, X_2 \geq 0, X_3 \geq 0.$$

ILLUSTRATION (6)

$$\begin{aligned}
\text{Max } Z &= 60\, X_1 + 70\, X_2 \\
\text{s. t.} \quad & X_1 + X_2 \leq 40 \\
& \tfrac{1}{3}X_1 + X_2 \leq 20
\end{aligned}$$

$$X_1 \geq 0, X_2 \geq 0.$$

ILLUSTRATION (7)

$$\begin{aligned}
\text{Min } Z &= 1.20\, X_1 + 2.00X_2 + 2.50X_3 + 3.00X_4 + 5.00X_5 + 6.00X_6 \\
\text{s. t.} \quad & 200\text{-}X_1 \leq (0.10)\,(200) \\
& 200\text{-}X_2 \leq (0.10)\,(200) \\
& 150\text{-}X_3 \leq (0.10)\,(150) \\
& 50\text{-}X_4 \leq (0.10)\,(50) \\
& 75\text{-}X_5 \leq (0.10)\,(75) \\
& 25\text{-}X_6 \leq (0.10)\,(25) \\
& 0.012X_1 + 0.014X_2 + 0.018X_3 + 0.040X_4 + 0.045X_5 + 0.060X_6 \leq 1.000 \\
& 0.010X_1 + 0.015X_2 + 0.018X_3 \leq 500 \\
& 0.025X_4 + 0.035X_5 + 0.040X_6 \leq 900
\end{aligned}$$

$$X_1 \geq 0, X_2 \geq 0, X_3 \geq 0, X_4 \geq 0, X_5 \geq 0, X_6 \geq 0,$$

ILLUSTRATION (8)

X_1 = lbs. of meat
X_2 = lbs. of bread
X_3 = lbs. of spinach

$$\begin{aligned}
\text{Min cost } Z &= 1.65\, X_1 + 0.70\, X_2 + 0.60\, X_3 \\
\text{s. t.} \quad & 40X_1 + 10X_2 + 5X_3 \geq 100 \\
& 8X_1 + 35X_2 + 6X_3 \geq 50 \\
& 5X_1 + 2X_2 + 20X_3 \geq 15 \\
& X_1, X_2, X_3 \geq 0
\end{aligned}$$

G. Linear Programming Module: Part 2 - Solution Algorithm Answers To Illustrative Exercises

ILLUSTRATION (10)

STEP 4

Z	X_1	X_2	X_3	X_4	X_5	RHS
1	−4	−3	−6	0	0	0
0	3	**1**	**3**	**1**	0	30
0	2	**2**	**3**	0	**1**	**40**

STEP 5

Z	X_1	X_2	X_3	X_4	X_5	RHS
1	−4	−3	−6	0	0	0
0	3	**1**	3	**1**	0	30
0	2	**2**	3	0	**1**	**40**

STEP 6

30/3 = 10

40/3 = $13\frac{1}{3}$

The **1st** constraint equation has the smallest RHS/X ratio.

STEP 8

(0 3 1 3 1 0 30) × ⅓ = (0 1 ⅓ 1 **⅓** 0 **10**)

STEP 9

(a) Multiply new pivot row by (**6**) and add to objective function row to change −6 of the pivot column to 0.

(b) Multiply new pivot row by (**−3**) and add to 2nd constraint equation row to change 3 of the pivot column to 0.

(c) 1 1 6 2 6 2

(0 1 ⅓ 1 ⅓ 0 10) × 6 =	(0 6 2 6 2 0 60)	add this to
objective function row:	(0 −4 −3 −6 0 0 0)	
new objective function row:	(0 2 −1 0 2 0 60)	

$$\begin{array}{ll}
\text{2nd constraint equation row:} & (0\quad 2\ \ 2\ \ 3\quad 0\ \ 1\ \ 40) \\
\text{2nd constraint eqn row:} & (0\ \ -1\ \ 1\ \ 0\ \ -1\ \ 1\ \ 10)
\end{array}$$

STEP 10

$10/(1/3) = (\mathbf{30})$

$(\mathbf{10/1}) = (\mathbf{10})$

STEP 11

The **2nd** constraint equation now has the smallest RHS/X ratio. New pivot element is 1.

(b) −1/3

$$\begin{array}{lcccccccl}
\textbf{(c)}\ (0\ -1\ 1\ 0\ -1\ 1\ 10) \times 1 = & (0 & -1 & 1 & 0 & -1 & 1 & 10 &) \\
\text{add this objective function row:} & (1 & 2 & -1 & 0 & 2 & 0 & 60 &) \\
\text{new objective function row:} & (0 & 1 & 0 & 0 & 1 & 1 & 70 &) \\
(0\ -1\ 1\ 0\ -1\ 1\ 10) \times (-1/3) = & (0 & 1/3 & -1/3 & 0 & 1/3 & -1/3 & -10/3 &) \\
\text{add this objective function row:} & (0 & 0 & 1/3 & 1 & 1/3 & 0 & 10 &) \\
\text{new 1st constraint equation row:} & (0 & 4/3 & 0 & 1 & 2/3 & -1/3 & 20/3 &)
\end{array}$$

STEP 12

$$\begin{array}{lll}
Z = 70 & X_2 = (\mathbf{10}) & X_3 = (\mathbf{20/3}) \\
X_1 = (\mathbf{0}) & X_4 = (\mathbf{0}) & X_5 = (\mathbf{0})
\end{array}$$

ILLUSTRATION (11)

$$\begin{array}{lllllll}
Z & -13\,X_1 & -10\,X_2 & & & & = 0 \\
 & X_1 & +\,X_2 & +\,X_3 & & & = 1{,}000 \\
 & X_1 & & & +X_4 & & = 600 \\
 & & X_2 & & & +X_5 & = 800
\end{array}$$

	Z	X_1	X_2	X_3	X_4	X_5	RHS
	1	−13	−10	0	0	0	0
	0	1	1	1	0	0	1,000
pivot row	0	1	0	0	1	0	600
	0	0	1	0	0	1	800
		pivot column					

New tableau after arithmetic procedures:

	Z	X_1	X_2	X_3	X_4	X_5	RHS
	1	0	−10	0	13	0	7,800
	0	0	1	1	−1	0	400
pivot row	0	1	0	0	1	0	600
	0	0	1	0	0	1	800

pivot
column

New tableau after arithmetic procedures:

Z	X_1	X_2	X_3	X_4	X_5	RHS
1	0	0	10	3	0	11,800
0	0	1	1	−1	0	400
0	1	0	0	1	0	600
0	0	0	−1	1	1	400

All in the objective function row are zero or positive, thus it is the optimal tableau.
Maximal profit of Z = 11,800 cents, with

$X_1 = 600$	produce 600 quarts of chocolate
$X_2 = 400$	produce 400 quarts of vanilla
$X_5 = 400$	slack; because we could sell up to 800 quarts of vanilla but we are only producing 400 quarts–there is a slack of 400 quarts when compared with maximum sale.

II. SOLUTIONS TO REGULAR PROBLEMS

Readers are expected to work out the solutions the regular problems in the end-of-the-chapter exercises. Accordingly, no solution is provided here.

III. SOLUTIONS TO SYNTHESIS EXERCISES AND PROBLEMS

For the synthesis exercises and problems, we are only providing solutions for selected problems. The reader is supposed to work out the missing problems by herself.

A. Remote Sensing and Geographic Information Systems

1. Bayesian Classifier

For this model, we used a 3 × 3 grid, each with an associated gray value (Wright and Chan 1994). Pixels 1, 2, 3, 6 and 9 belong to the pure-water class, with gray values averaging 2.6. Pixels 4, 5, 7 and 8 represented polluted water, with average gray-value of 7.75. Each **x** vector is made up of the row number of the pixel, the column number of the pixel, and the gray value for the pixel. The second-order surface is calculated, which successfully delineates the pure water from the polluted water.

To do this, we create a data-set consisting of nine **x** vectors, each containing a row number (the first entry x_1), a column number (the second entry x_2) and a value corresponding to the amount of ground-water pollution measured at that pixel (the third entry x_3). Thus the *k*th pixel is characterized by these three attributes: $\mathbf{x}_k = (x_{k1}\ x_{k2}\ x_{k3})^T$. Here pixels are ordered as they are in the text.

$$\begin{array}{lll}
\mathbf{x}_3 = (1\ 3\ 2)^{\mathrm{T}} & \mathbf{x}_6 = (2\ 3\ 3)^{\mathrm{T}} & \mathbf{x}_9 = (3\ 3\ 4)^{\mathrm{T}} \\
\mathbf{x}_2 = (1\ 2\ 3)^{\mathrm{T}} & \mathbf{x}_5 = (2\ 2\ 8)^{\mathrm{T}} & \mathbf{x}_8 = (3\ 2\ 7)^{\mathrm{T}} \\
\mathbf{x}_1 = (1\ 1\ 1)^{\mathrm{T}} & \mathbf{x}_4 = (2\ 1\ 7)^{\mathrm{T}} & \mathbf{x}_7 = (3\ 1\ 9)^{\mathrm{T}}
\end{array}$$

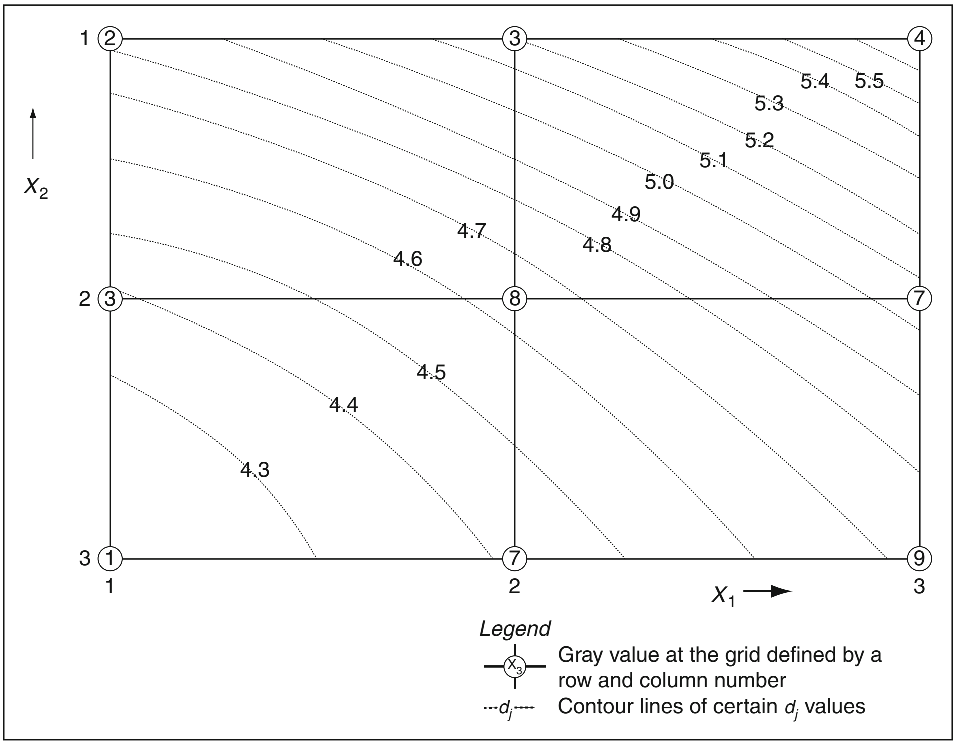
The number of pure-water pixels N_1 is 5, and the number of polluted-water pixels N_2 is 4. We calculate the mean vector and the covariance matrix. For example, the means are $\boldsymbol{\mu}_1 = (\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3 + \mathbf{x}_6 + \mathbf{x}_9)/\mathrm{N}_1$, and $\boldsymbol{\mu}_2 = (\mathbf{x}_4 + \mathbf{x}_5 + \mathbf{x}_7 + \mathbf{x}_8)/\mathrm{N}_2$. The covariance matrices are correspondingly $\mathbf{C}_1 = (\mathbf{x}_1\mathbf{x}_1^{\mathrm{T}} + \mathbf{x}_2\mathbf{x}_2^{\mathrm{T}} + \mathbf{x}_3\mathbf{x}_3^{\mathrm{T}} + \mathbf{x}_6\mathbf{x}_6^{\mathrm{T}} + \mathbf{x}_9\mathbf{x}_9^{\mathrm{T}})/\mathrm{N}_1 - \boldsymbol{\mu}_1\boldsymbol{\mu}_1$ and $\mathbf{C}_2 = (\mathbf{x}_4\mathbf{x}_4^{\mathrm{T}} + \mathbf{x}_5\mathbf{x}_5^{\mathrm{T}} + \mathbf{x}_7\mathbf{x}_7^{\mathrm{T}} + \mathbf{x}_8\mathbf{x}_8^{\mathrm{T}})/\mathrm{N}_2 - \boldsymbol{\mu}_2\boldsymbol{\mu}_2$. This results in $\boldsymbol{\mu}_1 = (1.6\ 2.4\ 2.6)^{\mathrm{T}}$, $\boldsymbol{\mu}_2 = (2.5\ 1.5\ 7.75)^{\mathrm{T}}$, $\mathbf{C}_1 = \begin{bmatrix} 0.64 & 0.36 & 0.64 \\ 0.36 & 0.64 & 0.56 \\ 0.64 & 0.56 & 1.04 \end{bmatrix}$, and $\mathbf{C}_2 = \begin{bmatrix} 0.25 & 0 & 0.125 \\ 0 & 0.25 & -0.125 \\ 0.125 & -0.125 & 0.688 \end{bmatrix}$. The inverse $\mathbf{C}_1^{-1}$ and $\mathbf{C}_2^{-1}$ are then taken. The Bayes decision-functions for pure water and polluted water are computed as $\mathrm{d}_1(x_1, x_2, x_3)$ and $\mathrm{d}_2(x_1, x_2, x_3)$,where $\mathrm{P}(\mathbf{z}_1)=5/9$ and $\mathrm{P}(\mathbf{z}_2) = 4/9$.

Solve the equation $\mathrm{d}_1(\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3) = \mathrm{d}_2(\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3)$ for x_3 in terms of x_1 and x_2 so that we can vary x_1 and x_2 across the region of interest and obtain a plot of x_3. The end results is a surface plot which should separate pure from polluted water. In Figure S.1 is the surface delineating the pure and polluted water—the pure water pixels all fall under the surface while the polluted water pixels all appear above the surface.

2. Weighted Iterative Conditional Mode (ICM) Algorithm

In performing the ICM algorithm, we noticed that first- and second-order neighbors are weighted the same when determining the allocation of a specific pixel (Wright and Chan 1994). But according to Tobler's first law, proximity is a factor during allocation. If we assume the distance between the pixel in question and its first-order neighbor on an ordinary square-grid is unity, the distance between that pixel and its second-order neighbors is √2. We follow an inverse relationship between distance and importance in determining the allocation of some central pixels (though an inverse squared or other relationship may have been used). We obtain the weights by scaling the above relationship so that sum of all neighbors of a pixel is still 8. This way, comparison may be made between this algorithm and the previous un-weighted one. The weights obtained are 1.1716 for first-order neighbors and 0.8284 for second-order neighbors. The sum over all of a pixel's neighbors is (4)(1.1716) + (4)(0.8284) = 8 and the first-order neighbors are 1.1716/0.8284 = √2 times as important as second-order neighbors.

Figure S.1 SURFACE SEPARATING PURE AND POLLUTED WATER



In the original ICM-algorithm, we simply scale the difference in the number of pixels assigned to each subregion and add it to 0.5. Here, we scale the weighted difference in the number of pixels and add it to 0.5. For instance, if a pixel has two first-order neighbors assigned to subregion 1 and two assigned to subregion 2, and two second-order neighbors assigned to each as well, the context makes no difference. The reason is that the weighted sum is 4—meaning that 0 is added to 0.5 to constitute the 'compare' value. In the extreme case, however, four first-order neighbors allocated to subregion 1 and zero second-order neighbors allocated to the same subregion would, in the un-weighted algorithm, result in the same determination: context makes no difference. However, with weighting, the sum is 4.6864 and the algorithm will increase the probability that the pixel belongs to subregion 1 (the exact amount depends upon β and σ). Besides the weighting, the algorithm works in the same way.

We obtain the initial mean of the entire data-set gray-values—$\mu = 4.722$ and $\sigma = 1.981$—and standardize the grey values to N(0.5, 1). Anything under 0.5 is an initial guess of clean pixels, anything over is the polluted pixels. Based on initial guesses, we calculate means and standard deviations of standardized grey-values for each initial subset, and obtained $\mu_1(\mathbf{x}) = -0.243$, $\sigma_1^2(\mathbf{x}) = 0.227$, $\mu_2(\mathbf{x}) = 1.739$, $\sigma_2^2(\mathbf{x}) = 0.446$. Now we set up the iteration across the eligible pixels (no edges included) using the ICM equation, with $\beta = 1.14$.

Implementation of the ICM can be seen to be almost identical in both the weighted and un-weighted cases. Difference lies in the calculation of the "compare" values, in which the summation must be broken into a first-order and a second-order summation. We were able to obtain the same delineation of polluted and unpolluted ground-water using a lower β of 1.14 (from 1.34 for the un-weighted case)—as alluded to above. The upper boundary for obtaining the same delineation also improved to $\beta = 1.151$. Above this β, the upper right-corner was allocated to the unpolluted subregion. Convergence was obtained in two iterations.

This result shows that the weighted ICM can be used to allow β to be more flexible in obtaining "correct" pixel allocation than the un-weighted case. Also, weighting the second-order neighbors less than the first-order neighbors seems a more realistic formulation of the problem and could be extended to more than two orders of neighbors. In short, advantages of the weighted ICM over the un-weighted ICM include the following: (a) more flexibility in β value selections and (b) a more realistic representation of how different order-neighbors affect a pixel's allocation. The weighted ICM-algorithm appears to be a promising tool for use in delineating polluted from unpolluted ground-water. Further areas of improvement include deciding the correct weight to use and extending the model to three dimensions. If non-homogeneity of the ground composition results in irregular-shaped pollution-plumes, the weighted ICM-algorithm should have no problems with classification since the global approximation of the ground truth is comprised of the collection of each pixel's local truth.

3. Combined Classification Scheme

(*a*) *Weighted ICM algorithm*. In this weighted ICM algorithm, the procedure is similar to that described in Section 1.c and will not be repeated here (Wright and Chan 1994a). However, we wish to emphasize once again that with weighting—as contrasted with the un-weighted case—the "compare"-value step works differently. Here, the difference between $\hat{T}_{1j}$ and $\hat{T}_{2j}$ is 1.3728 and the algorithm will increase the probability that the pixel belongs to subregion 1 (the exact amount depends upon β and σ).

We have also implemented the algorithm for edge pixels. These edge pixels do not have enough neighbors to sum to eight, but the relative importance between first- and second-order neighbors remains the same. The only difference with these pixels is that, since nothing is known about the "other side" falling beyond the boundary, the possible amount of contextuality applied is less than that for interior points.

(*b*) *MCDM formulation*. The X space in this problem consists of a binary decision-variable representing each pixel. In the context of this problem, turning the variable for a pixel "on" ($x_i = 1$) implies delineating that pixel as polluted, whereas leaving it "off" ($x_i = 0$) implies an unpolluted pixel. The X space for the sample problem here (a 10×10 grid) has 1.27×10^{30} discrete possibilities.

There are several possibilities in choosing the Y' space for this problem. First, we considered varying the ranges of β and the variance applied when creating the standardized grey-values in the algorithm. This method looks promising to obtain a model with a large amount of flexibility, since it is preferable to maximize the ranges of each. Due to the possible ranges of σ being so small, however, we decided against this alternative.

The Y' space presented in this project pits the choice of the channel-1 weight (and by default, then, the channel-2 weight as well) against the choice of β. This is reasonable since we would like to minimize the value of β in order to "let the data speak for themselves" as much as possible.

(*c*) *Noninferior solutions*. The first step is to classify the pixels into polluted and unpolluted subregions. To do this, we calculate the mean of all the pixels. Any values below this mean are considered unpolluted, any values above this mean are considered polluted. Next, we standardized those pixels initially allocated to the unpolluted subregion to a mean of 0 and a standard deviation of 0.25. We also standardize those pixels initially allocated to the polluted subregion to a mean of 1 and a standard deviation of 0.25. Even though we could have chosen from a range of possible standard deviations, we chose 0.25 since two standard deviations above the unpolluted mean and two standard deviations below the polluted mean is 0.5. That is, there is only a 2.5 percent overlap between the two subregions distributions.

Then, we determine for each pixel a value based on the number of neighbors in each subregion, σ^2, and β. The standardized grey-value is compared against this value; if the standardized grey-value is less than the value, the pixel is allocated to the unpolluted subregion, otherwise it is allocated to the polluted subregion. These new allocations are then used as input for the next iteration. In most problems, convergence occurs after one iteration. We used 0 through 1 in increments of 0.1 for the channel-1 weights (channel-2 weight equals one minus channel-1 weight). We also stepped through β in whole-number increments from 0 (no contextuality) to 10. The feasible Y′ space in this problem consists of all positive β-values and all channel-1 weights between 0 and 1 inclusive.

The preference structure we selected for this problem consists of a specific allocation of pixels as seen above in the "ground truth." The smallest "distance" from that ground truth represents a non-dominated solution. Zero "distance" is considered Pareto optimal. We consider the number of pixels in the ICM-generated solution different from the ground truth to be this "distance."

A contour plot of the "distances" away from the ground truth is shown in Figure S.2 The results from any preference structure could have been displayed, so this

Figure S.2 CONTOUR OF ERRED PIXELS FROM THE GROUND TRUTH

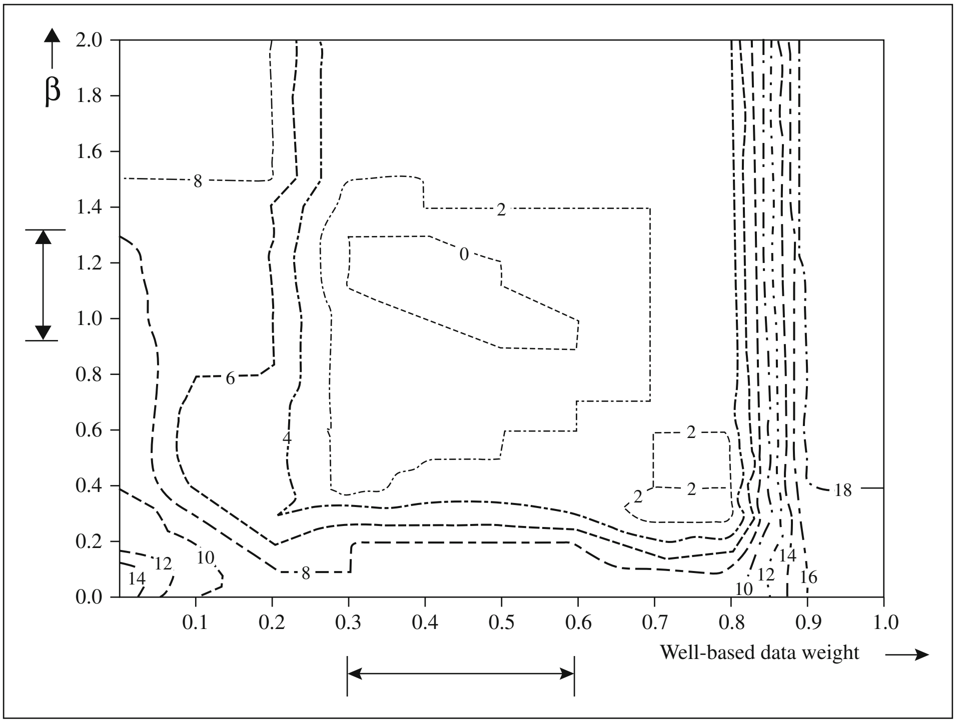
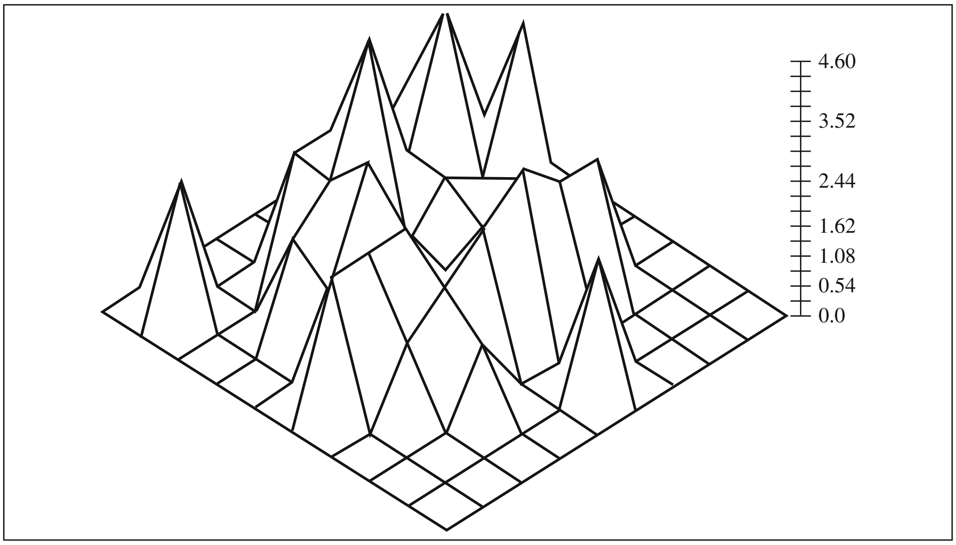


Figure S.3 IMAGE FROM WEIGHTED CHANNELS



plot is not unique. There is a contour area within which the Pareto optimal was obtained, all within the β range of 0.9 to 1.3 with a channel-1 weight ranging from 0.3 to 0.6. Since the result is Pareto optimal, translation back to the X space results in the same plot as the "ground truth" plot. Figure S.3 shows a plot of the weighted channels using channel-1 weight equal to 0.4 and channel-2 weight equal to 0.6. Compared with the original "ground truth," we see the pitting found in channel 2 less severe, but still see some noise around the perimeter of the plot. Thus $\beta = 3$ is necessary to be greater than 0 (non-contextual) in order to delineate these noisy pixels as unpolluted areas.

B. Facility Location

1. Quadratic-Assignment Problem

(a) *Formulation*. The problem is formulated as the following linear binary program. A, B, C, D denote the four work-stations, which are to be placed in locations a, b, c, d. When workstation A is located at a, XAa = 1, etc.

MIN
27200YABab + 25600YABac + 32000YABad + 28800YABbc + 16000YABbd
+ 14400YABcd + 13600YACab + 12800YACac + 16000YACad + 8000YACbd
+ 7200YACcd + 10200YADab + 9600YADac + 6000YADbd + 540YBDcd
+ 6800YBDab + 6400YBDac + 8000YBDad + 7200YBDbc + 4000YBDbd
+ 3600YBDcd + 3400YCDac + 3200YCDac + 4000YCDad + 3600YCDbc
+ 2000YCDbd + 1800YCDcd

s.t.
XAa+XAb+XAc+XAd=1
XBa+XBb+XBc+XBd=1
XCa+XCb+XCc+XCd=1
Xda+XDb+XDc+XDd=1
XAa+XBa+XCa+XDa=1
XBb+XCc−2YBCbc>=0
XBb+XCd−2YBCbd>=0
XBc+XCd−2YBCcd>=0
XBa+XDb−2YBDab>=0
XBa+XDc−2YBDac>=0
XAa+XCd−YACad<=1
XAb+XCc−YACbc<=1
XAb+XCd−YACbd<=1
XAc+XCd−YACcd<=1
XAa+XDb−YADab<=1

```
XAb+XBb+XCb+XDb=1
XAc+XBc+XCc+XDc=1
Xad+XBd+XCd+XDd=1
XAa+XBb−2YABab>=0
XAa+XBc−2YABac>=0
XAa+XBd−2YABad>=0
XAb+XBc−2YABbc>=0
XAb+XBd−2YABbd>=0
XAc+XBd−2YABcd>=0
XAa+XCb−2YACab>=0
XAa+XCc−2YACac>=0
XAa+XCd−2YACad>=0
XAb+XCc−2YACbc>=0
XAb+XCd−2YACbd>=0
XAc+XCd−2YACcd>=0
XAa+XDb−2YADab>=0
XAa+XDc−2YADac>=0
Xaa+XDd−2YADad>=0
XBa+XCb−2YBCab>=0
XBa+XCc−2YBCac>=0
XBa+XCd−2YBCad>=0
XBa+XDd−2YBDad>=0
XBb+XDc−2YBDbc>=0
XBb+XDd−2YBDbd>=0
XBc+XDd−2YBDcd>=0
XCa+XDb−2YCDab>=0
XCa+XDc−2YCDac>=0
XCa+XDd−2YCDad>=0
XAb+XDc−2YADbc>=0
XAb+XDd−2YADbd>=0
Xac+XDd−2YADcd>=0
XCd+XDc−2YCDbc>=0
XCb+Xdd−2YCDbd>=0
Xcc+XDd−2YCDcd>=0
XAa+XBb−YABab<=1
XAa+XBc−YABac<=1
XAa+XBd−YABad<=1
XAb+XBc−YABbc<=1
XAb+XBd−YABbd<=1
XAc+XBd−YABcd<=1
XAa+XCb−YACab<=1
XAa+XCc−YACac<=1
XAa+XDc−YADac<=1
XAa+XDd−YADad<=1
XAb+XDc−YADbc<=1
XAb+XDd−YADbd<=1
Xac+XDd−YADcd<=1
XBa+XCb−YBCab<=1
XBa+XCc−YBCac<=1
XBa+XCd−YBCad<=1
XBb+XCc−YBCbc<=1
XBb+XCd−YBCbd<=1
XBc+XCd−YBCcd<=1
XBa+XDb−YBDab<=1
XBa+XDc−YBDac<=1
XBa+XDd−YBDad<=1
XBb+XDc−YBDbc<=1
Xbb+XDd−YBDbd<=1
XBc+XDd−YBDcd<=1
XCa+XDb−YCDab<=1
XCa+XDc−YCDac<=1
XCa+XDd−YCDad<=1
XCd+XDc−YCDbc<=1
XCb+Xdd−YCDbd<=1
XCc+XDd−YCDcd<=1
```

(*b*) *Solution*. Solution yields an objective function of zero and the following unitary-valued variables XAd, XBc, XCb, XDa; with the rest of the variables assuming zero. In other words, work-station A is assigned to location d, workstation B to c, C to b and D to a.

(*c*) *Discussion*. While the binary variables are in agreement with the answer given in the text, the zero objective-function-value may look 'strange' at first sight. If the linear version is merely an approximation of the nonlinear model, one would expect the objective function to reflect the minimum cost of assignment and be other than zero. A moment's reflection, however, would reveal that zero is the correct answer, and the objective function of the linear model is not an approximation of the nonlinear objective. In the notation of the quadratic assignment model in Section VI of Chapter 4, a zero linear-objective merely reflects that all the $y_{klij} = x_{ki}x_{lj} = 0$—hence the objective function that sums over all y_{klij} is correspondingly zero. This says that the variables x_{ki} and x_{lj} cannot be unitary-valued simultaneously. Only one of the two can be unity at most, with the other being zero. In other words, the assignment of work-station *k* to location *i* and the assignment of work-station *l* to location *j* cannot take place simultaneously. This is reinforced by the observation that the constraints $x_{ki} + x_{lj} - 2y_{klij} \geq 0$ are non-binding, suggesting a degenerate case.

C. Location-Routing

1. Districting

The application of the location model requires several steps as described in statement of the problem (Patterson 1995). We start with *Phase I* of the algorithm, which does the partitioning.

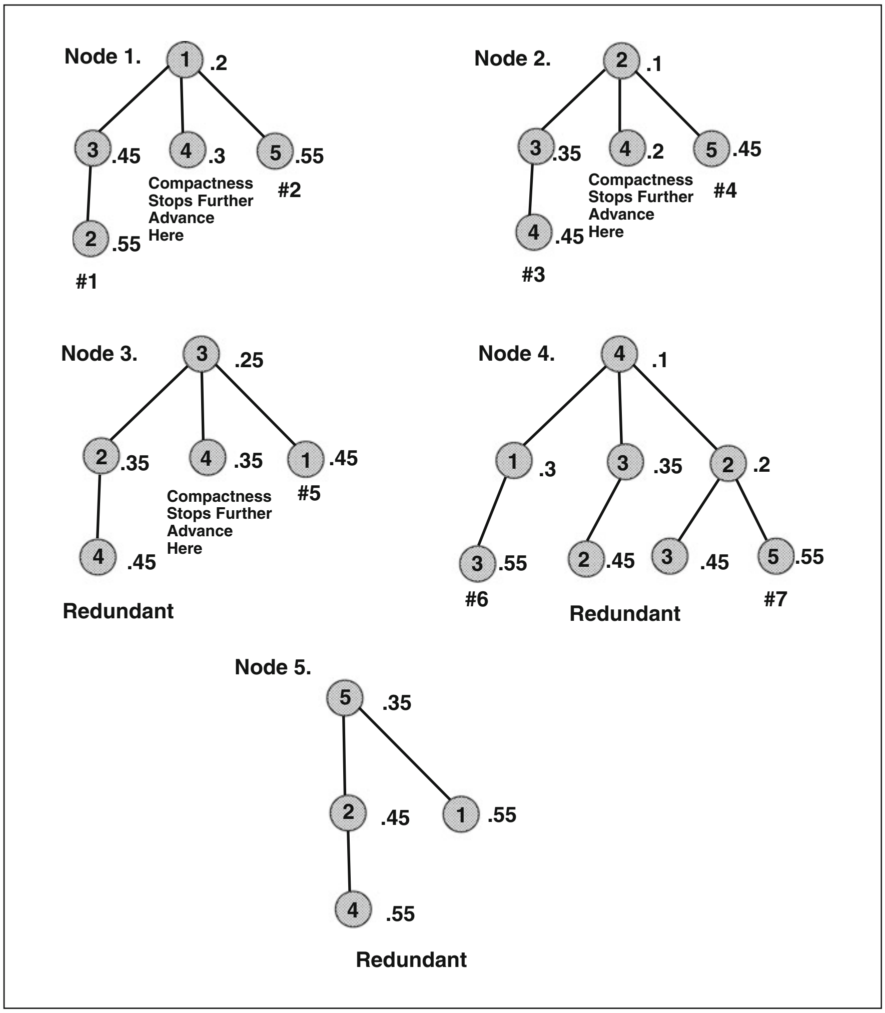
Step 1.1: Choose the number (p) of maintenance depots to be located within the communications network. We choose $p = 2$. Therefore $1/p = 0.50$, and the tolerance (α) is set to 0.10 as suggested.

Step 1.2: Partitioning algorithm. This algorithm is again broken into two steps. *Phase I* is a complete enumeration of the possible subnetworks. The maximum distance for compactness is arbitrarily set at 150 (which is arbitrarily chosen to show the elimination of a possible subnetwork because of proximity concerns, that being a subnetwork consisting of nodes 1-2-4). The Phase I subnetwork set is shown in Figure S.4.

Phase II implements the model described in the problem statement. The model is formulated and solved using an EXCEL spreadsheet. The formulation

Partitioning Phase I

Figure S.4 TREE-SEARCH PARTITIONING ALGORITHM



SOURCE: Patterson (1995). Reprinted with Permission.

and solution are shown in the original problem statement. The results of this application are two subnetworks in which the maintenance depots are to be located. The best partition possible consists of nodes 1-3 and 2-4-5. This means that one depot is to be located at node 1 or 3, while the other is located at node 2, 4, or 5.

2. Minkowski's *Metric*

Given $\mathbf{y}^1 = (14, 13)$, $\mathbf{y}^2 = (4, 4)$, and $r'(\mathbf{y}; p) = [\Sigma_i | y_i^1 - y_i^2 |^p]^{1/p}$ as plotted in Figure S.5, we will sketch the various shapes of the Minkowski's metric.

(*a*) When $1 \le p \le \infty$, $r'(\text{y}; p)$ is called the l_p-metric. As $p \to \infty$ $r'(\mathbf{y}; p) =$ Max $\{|14-4|, |13-4|\} = 10$. As $p \to 1$, $r'(\mathbf{y}; p) = |14-4| + |13-4| = 19$.

(*b*) The Minkowski's metric is a generalization of l_p-metric, when p goes below unity in value. As $p \to 0$, $r'(\mathbf{y}; p) = \{|14-4|^o + |13-4|^o\}^\infty = \infty$; and as $p \to 1$, $r'(\mathbf{y}; p) = \{|14-4| + |13-4|\} = 19$.

For example, when $p = -1/2$, $r'(\mathbf{y}; -1/2) = \{|14-4|^{-1/2} + |13-4|^{-1/2}\}^{-2} = 2.37$. When $p = -1$, $r'(\mathbf{y}; -1) = \{|14-4|^{-1} + |13-4|^{-1}\}^{-1} = 4.74$. When $p = -2$, $r'(\text{y}; -2) = \{|14-4|^{-2} + |13-4|^{-2}\}^{-1/2} = 6.69$. As $p \to -\infty$, $r'(\mathbf{y}; p) \to$ Min $[|14-4|, |13-4|] = 9$.

Figure S.5 PLOT OF MINKOWSKI'S METRIC

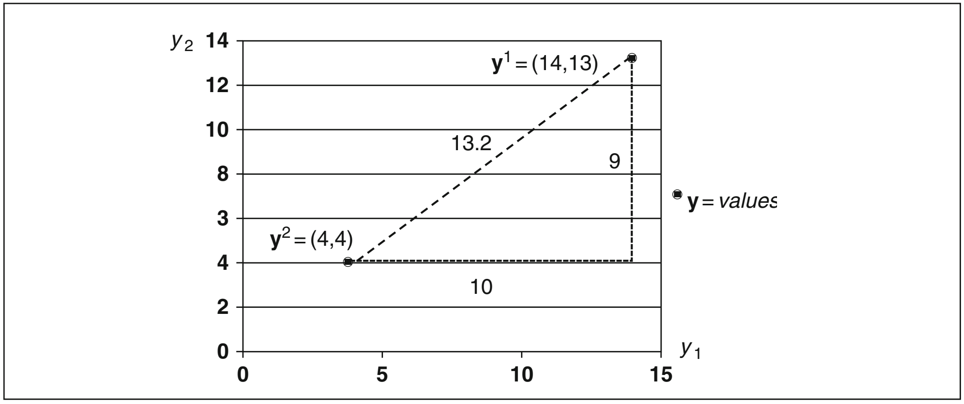


Figure S.6 METRIC AS FUNCTION OF p

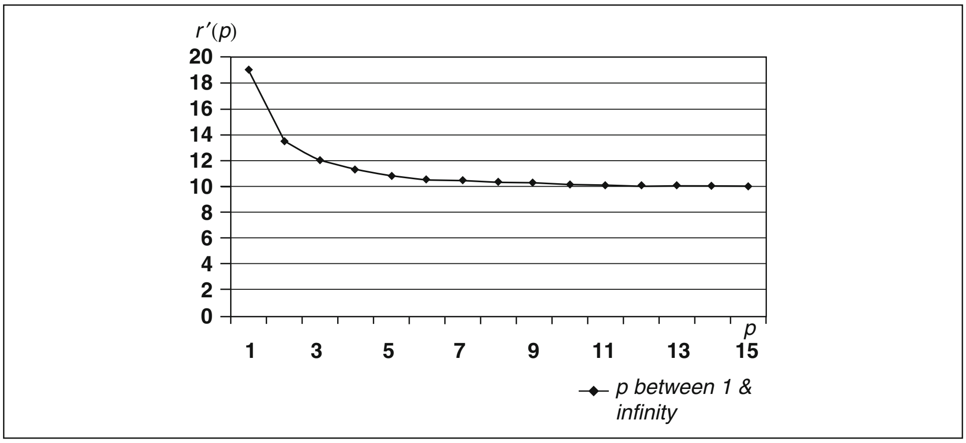


Figure S.7 METRIC WHEN p IS LESS THAN UNITY

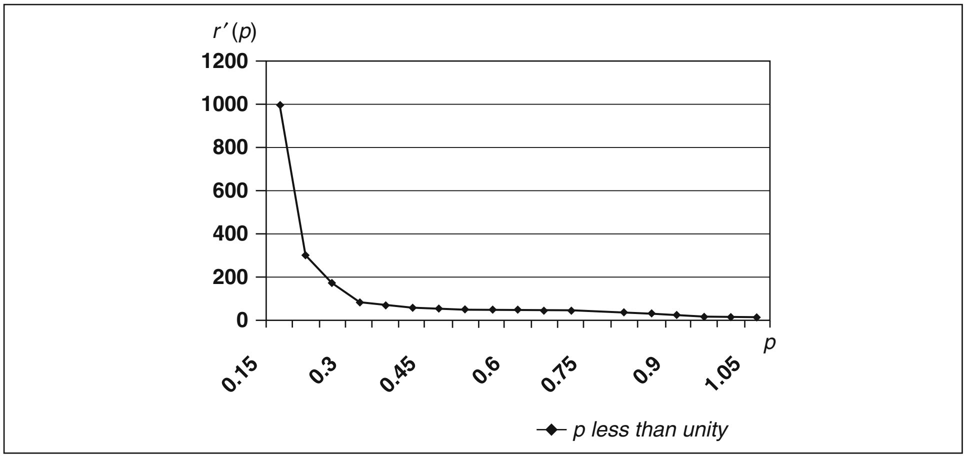
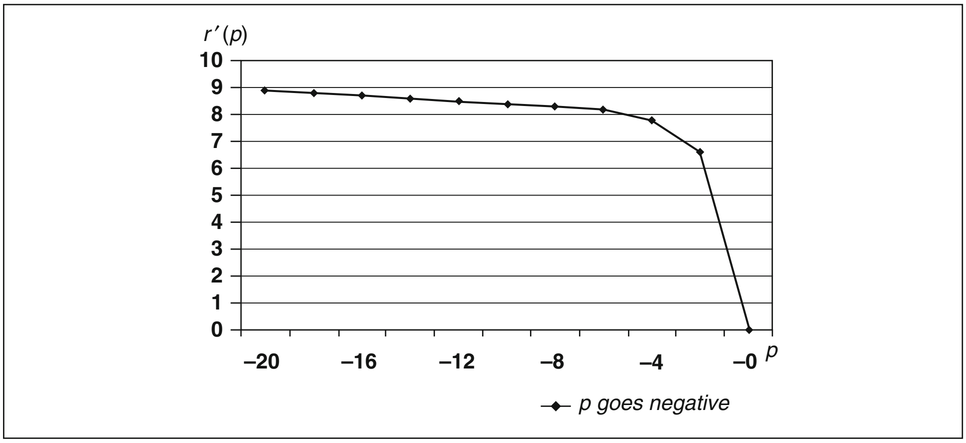


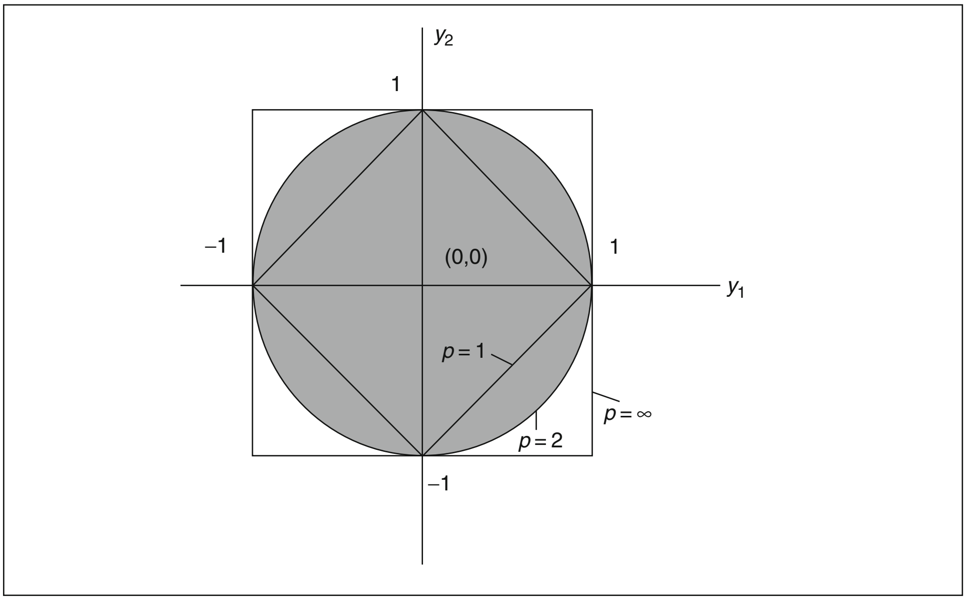
Figure S.8 METRIC AS p GOES NEGATIVE



(*c*) As $p \to \infty$, $r'(\mathbf{y}; p) \to$ Max $\{|14-4|, |13-4|\} = 10$. Now if we minimize r', representing say the distance to the ideal, then we are minimizing the maximum deviation from the ideal. Such a metric is often used to model situations where one minimizes the maximum regret.

(*d*) Refer to the plot in Figure S.9. The unit contours are defined as $(|y_1|^p + |y_1|^p)^{1/p} = 1$. It is clear that for $\mathbf{y}_1 = (0, 0)$ and $y_2 = y$, we have the following norms: $r' = (\mathbf{y}; 1) = (|y_1| + |y_2|) = 1$; $r'(\mathbf{y}; 2) = (|y_1|^2 + |y_2|^2)^{1/2} = 1$; and r' $(\mathbf{y}; \infty) = \lim_{p\to\infty} (|y_1|^p + |y_2|^p)^{1/p} =$ Max $\{|y_1|, |y_2|\} = 1$ meaning that either $|y_1| = 1$ or $|y_2| = 1$. Normalized to be unity in value, $r'(\mathbf{y}; 1)$ shows a totally-compensatory utility-function of $\mathbf{y} = (|y_1|, |y_2|)$, i.e., there is an exact tradeoff between $|y_1|$ and $|y_2|$. On the other hand, $r'(\mathbf{y}; \infty)$ denotes a totally-noncompensatory utility function, i.e., either $|y_1|$ or $|y_2|$ would prevail, depending on which one is larger. $r'(\mathbf{y}; 2)$ is somewhere in between—neither totally-compensatory nor totally-noncompensatory. In general, the higher the value of p the more weight is given the attribute which is larger.

Figure S.9 ISO-CONTOURS OF UNITY FOR l_p-METRIC



(*e*) Notice that when there is only one attribute y_1 or y_2, the three functions are identical, since there is no tradeoff between two attributes anymore. Thus the points (1,0),(0,1),(−1,0),(0,−1) are always the same regardless of *p* value.

D. Activity Derivation, Competition and Allocation

1. Multicriteria Game

In a zero-sum game, DM1 (playing 'offensive') maximizes his minimum gain while DM2 (playing 'defensive') minimizes his maximum loss, and the gain of DM1 is identical to the loss of DM 2 (Zelany 1982). Multiple payoff is in terms of a vector (rather than a scalar):

		DM2		
		q'_1	q'_2	q'_3
DM1	p'_1	(3,2)	(3,4)	(1,5)
	p'_2	(2,1)	(3,2)	(2,2)
	p'_3	(4,1)	(1,3)	(3,1)

Thus if both DMs decide to play their second option, DM1 wins 3 units in the first dimension and 2 in the second. DM2 loses the same amounts. p'_{i} and q'_{j} denote the probability DM1 and DM2 will play the *i*th and *j*th strategy respectively. A pure strategy is when *p*'s and *q*'s are 1 or 0 in value.

Each vector payoff $a_{ij} = (a^1_{\mathrm{ij}}, a^2_{\mathrm{ij}})$ is to be replaced by a convex combina tion of both components: $wa^1_{\mathrm{ij}} + (1–w)a^2_{\mathrm{ij}}$. For example, $a_{11} = 3w + (1 - w)2 = w + 2$, and so on. It can be shown that an LP can be set up to solve this problem if variables p and q are defined such that $p' = pz'$ and $q' = qz'$ and $z' = 1/z$. Notice that $p_1 + p_2 + p_3$ is not necessarily unity in this case, neither would $q_1 + q_2 + q_3$. Now we have the game for DM2 as follows:

$$\begin{array}{ll} & \text{Max } z = q_1 + q_2 + q_3 \\ & (w+2)q_1 + (4-w)q_2 + (5-4w)q_3 \le 1 \\ \text{s.t.} & (w+1)q_1 + (w+2)q_2 + 2q_3 \le 1. \\ & (3w+1)q_1 + (3-2w)q_2 + (2w+1)q_3 \le 1 \end{array}$$

The initial tableau is given by

	q_1	q_2	q_3	q_4	q_5	q_6	RHS
	-1	-1	-1	0	0	0	0
q_1	w+2	4$-w$	5$-$4w	1	0	0	1
q_2	w+1	w+2	2	0	1	0	1
q_3	3w+1	3$-$2w	2w+1	0	0	1	1

(Notice it will be a minimization LP for DM1)

Set $w = 0$ and solve by simplex. Then explore the optimality for parameter w changing from 0 to 1.

For $0 \le w \le 3/5$ the optimal solution is

$$q_1 = \frac{1}{2 + W}, \ q_2 = 0, \ q_3 = 0$$

and the dual solution is

$$p_1 = \frac{1}{2+w}, p_2 = 0, p_3 = 0$$

while the objective function reaches $1/(2+w)$. For $w \ge 3/5$ the optimal solution is

$$q_1 = 0, q_2 = 0, q_3 = \frac{1}{5-4w}$$

and the dual solution is

$$p_1 = \frac{1}{5-4w}, p_2 = 0, p_3 = 0$$

while the objective function reaches $1/(5 - 4w)$.

It appears that there are two non-dominated pairs of pure strategies:

(i) If w is between 0 and 3/5,

$$(q'_{1,} q'_{2,} q'_3) = (2+w)\left(\frac{1}{2+w}, 0, 0\right) = (1,0,0)$$

and

$$(p'_{1,}p'_{2,}p'_3) = (2+w)\left(\frac{1}{2+w}, 0, 0\right) = (1, 0, 0).$$

(ii) If *w* is at or bigger than 3/5, then

$$(q'_{1,}q'_{2,}q'_3) = (5-4w)\left(0, 0, \frac{1}{5-4w}\right) = (0, 0, 1)$$

and

$$(p'_{1,}p'_{2,}p'_3) = (5-4w)\left(\frac{1}{5-4w}, 0, 0\right) = (1, 0, 0).$$

The respective payoffs are (3, 2) and (1, 5) as shown in the payoff matrix under row 1 and column 1 for case (i) and row 3 column 1 in case (ii).

Depending on *w*, an average payoff can be calculated. Observe that for $w = 3/5$ both strategies lead to the same average return: $3/5(3) + 2/5(2) = 3/5(1) + 2/5(5) = 2.60$. Patterson et al. (1994) showed that multiple optimal-solutions exist at $w = 0.6$, consisting of both pure and mixed strategies. For $0.6 \leq w \leq 1$ mixed strategies exist in addition to the pure strategy. The multiple solution is consistent with the transition from pure to mixed strategies as the value of *w* crosses the 0.6 mark. Furthermore at $w = 1$, it can be shown that value of the game is 2.333, representing another local maximum for this game. This 'second' maximum corresponds to the mixed strategy of $p'_1 = 0$, $p'_2 = 0.67$, and $p'_3 = 0.33$ ($q'_1 = 0$, $q'_2 = 0.333$, and $q'_3 = 0.667$).

2. Gravity vs. Transportation Model

(*a*) When $\alpha = 0$, $C_{ij}^{-\alpha} = 1$ and assumes its maximum value.

(*b*) When $\alpha = \infty$, $C_{ij}^{-\alpha} \to 0$, *assuming its minimum value.*

(*c*) The equation $z'_{ij} = C_{ij}V_{ij}$ becomes

$$V_{ij} = (z'_{ij})(1) \quad or \quad z'_{ij} = V_{ij}$$

$$z = \Sigma_{ij}\, z'_{ij} = \Sigma_{ij}\, V_{ij}$$

with the same constraints. The minimization solution will be indeterminate among all trip-distribution feasible-solutions.

(*d*) The equation $z'_{ij} = C_{ij}V_{ij}$ becomes very small with $C_{ij}^{-\alpha} \to 0$, with $z'_{ij} \to \infty$ for any finite V_{ij}. The objective function is a strong driving force in determining the resulting trip-distributions $V_{ij}*$.

(*e*) Part (c) suggests a trip distribution independent of travel cost while part (d) suggests one that is particularly sensitive to travel cost.

3. Calibration of a Doubly-Constrained Model

The calibration equations are shown in Chapter 3, Section B. Notice the two equation-sets are coupled together, in that k appears on the right-hand-side of the first equation set, and l appears on the right-hand-side of the second. An iterative solution strategy is anticipated.

We wish to solve the four equations and four unknowns for k_1, k_2, l_1, and l_2 as represented by Equation 3.47 when $n' = 2$:

$$k_1 = (400l_1+62.5l_2)^{-1} \qquad k_2 = (100l_1+250l_2)^{-1}$$
$$l_1 = (300k_1+87.5k_2)^{-1} \qquad l_2 = (75k_1+350k_2)^{-1}$$

Suppose we start with the arbitrary values of 1 for the k's. Substituting 1's in the formulas will yield $l_1 = 0.00258$, $l_2 = 0.00235$. Now substitute these l values into the formulas for the k's in the above equation set, one will find that these new values for the k's are no longer 1's. We continue this process until a consistent set of k's and l's are obtained, as shown in the Table below. It can be seen that we obtain convergence within five iterations.

Iteration	k_1	k_2	l_1	l_2
1	1	1	0.00258	0.00235
2	0.84827	1.18273	0.00279	0.00209
3	0.80217	1.24766	0.00286	0.00201
4	0.78763	1.26823	0.00288	0.00199
5	0.78347	1.27307	0.00289	0.00198
6	0.7810	1.27551	–	–
Final	0.781	1.276	0.0029	0.0020

E. Land-Use Models

1. Economic Base and Activity Allocation

Let us examine Chan and Rasmussen (1979) in terms of both the aggregate total-forecast and the subareal housing-distribution. This is to verify the soundness of the Economic-base Theory and the Gravity Spatial-interactance relationship—two of the basic building blocks of many urban-development models. The Chan and Rasmussen study appears to yield lower forecasts in general than the CRPC approach. This is attributed to the fact that the Chan and Rasmussen procedure considers not only the demand for housing, but also the economic base to support new residents in the area. Furthermore, the actual housing requirement is tempered by the availability of housing supply. The CRPC approach, on the other hand, does not seem to address the problem of land-holding capacity. It is a "statement of the need" for additional housing, whether or not land is actually available for new housing development. The former can be view as the realizable demand while the latter can be interpreted as the gross demand.

Commuting between home and the place of employment is recognized by the Chan and Rasmussen study as one of the major determinants of residential location. As such, spatial interaction is explicitly modeled by a gravity-type formulation that locates residents in relation to their place of employment.

The CRPC study, on the other hand, is a good deal less specific in dealing with locational choice, where subareal housing is simply derived from its population projection. In other words, while the Chan and Rasmussen study recognizes the coupling relationship between transportation and land-use, less emphasis is given by the CRPC housing forecast. It is not surprising, therefore, that the Chan and Rasmussen study projects (quite realistically) more clustering of high-density housing close to State College, which is by far the largest employment center of the region.

Both the CRPC and Chan/Rasmussen study assume that there are no substantial in-or-out migration, suggesting the student enrollment at Penn State would stabilize at 31, 500 by 1985. Both studies again assume the existing trends, including birth/death rates and other coefficients and ratios, will remain constant over time for each township. These assumptions, particularly the first two, did not hold true over the years. The scientific resources at Penn State University have attracted new industries (and therefore population) into the region. Defying the demographic projection, participation of a more mature student-body broke the enrollment ceiling forecasted for the traditional, post-World-War-II 18–21 age-group. Over the ten years from 1975 to 1985, State College and its immediate environs have decidedly gotten more urban than anticipated. This is evidenced by the unexpected increase in multiple-family units in State College and all the townships in the Center Region. In State College, single-family units are replaced by multiple-family units. With the exception of a decline in State-College proper, single-family units also increase elsewhere to a level comparable to the CRPC forecast, which is above the Chan and Rasmussen study. In short, the observed housing-units are closer to the CRPC optimistic-forecast than the Chan and Rasmussen study. The significant in-migration makes the difference, suggesting that there are really other "basic industries" beyond higher education—a fact overlooked by Chan and Rasmussen.

F. Spatial-Temporal Information

1. Cohort-Survival Method

(*a*) Crude birthrate of any region is defined as birth-per-person (or per 100 persons) in that period for that region (Jha 1972). For example, if the number of births for York County is 2000 for five years (1940–1945) and its average population over the period is 210, 000, crude birthrate for York in 1940–1945 is 2,000/210,000 = 1/105, or we can say that crude birthrate is one in 105 people.

Crude death-rate is similar to crude birthrate. If in the above example for the same period, the number of deaths for different reasons is 500, then crude death-rate for that period will be 500/210,000 = 0.0023.

If the total number of people coming into York County in the five-year period (1940–1945) is 1,400 and the outgoing number from this region is 1,295, then net migration will be 1,400–1,295 = 105. This will, of course, be net in-migration.

(*b*) From the given Table, the number of children born in 1940–1945 is 7. The probability of having a child for age-group 15–19 is [7/(14 + 16 + 21)] [14/(14 + 16 + 21)] = (0.137)(0.275) = 0.035. Here (14 + 16 + 21) = 51 is the total number of women of child-bearing age in 1945, ranging from 15 to 29 year old.

The probability of having a child for the woman age-group 20–24 is (0.137)(16/51) = 0.043. The probability of having a child for the woman age-group 25–29 is (0.137)(21/51) = 0.0565. For the desired growth-matrix, the top row of the matrix—the birth rates $\bar{\mathbf{b}}^T = (\leftarrow b_{ij} \rightarrow)$—may be calculated as follows:

$$b_{13} = \left[\frac{b(10-14)}{2} + \frac{s_5 + \cdots + s_9}{s_0 + \cdots + s_4} \frac{b(15-19)}{2} \right] \frac{s_0 + \cdots + s_4}{s(0)} \frac{N_f}{N_c}$$

and so on. Here $b(10-14)$ is the fertility-rate of age-group 10–14 (which is zero in our case), $s_{ij} = (s_5 + \cdots + s_9)/(s_0 + \cdots + s_4)$ is the surviving ratio of cohort-group i (0–4 year) in group j (5–9 year), N_f is the number of female children, while N_c is the total number of children. As a degenerate case for the postnatal population, the probability of living from birth to the end of five-years is $(s_0 + \cdots + s_4)/s(0)$, where $s(0)$ is the number of births to begin with.

We can now write the equation $\mathbf{G}\,\mathbf{N}(t) = \mathbf{N}(t + \Delta t)$ as discussed in the "Interregional growth and distribution" section of the "Economics" chapter, where $\mathbf{G}$ is the 6×6 growth-matrix, $\mathbf{N}(t)$ is the female-population in 1940, and $\mathbf{N}(t + \Delta t)$ is the female-population in 1945. Here the surviving-ratio $(s_0 + \cdots + s_4)/s(0)$ is given as 0.98 and the percentage of female-children is $N_f/N_c = 0.49$. Over five-years, we have (5)(0.98)(0.49) = 2.43 times as many female children, following the last part of the above birthrate equation: $[(s_0 + \cdots + s_4)/s(0)](N_f/N_c)$.

In the absence of migration, the sub-diagonal elements of the growth-matrix $\mathbf{G} = [G_{ij}]$ can be calculated as follows. Starting with the surviving-ratio $(s_5 + \cdots + s_9)/(s_0 + \cdots + s_4)$ for the 5–9 year group, where the numerator is 10 and the denominator is also 10, the ratio is unity. Here are the detailed calculations for the remaining age-groups:

Age	0–4	5–9	10–14	15–19	20–24	25–29
Group (i)	1	2	3	4	5	6
calculation	1	12/14	14/15	16/18	21/22	–
$S_{i+1,i} = G_{i+1,i}$	1	0.86	0.93	0.89	0.95	–

Elements of the top row of the growth matrix, or the birthrate for female-children b_{ij}, are computed from the birthrate equation as follows:

$$b_{13} = [(0.86)(35/2)(2.43)]/1000 = 0.036$$
$$b_{14} = [(32/2)+(0.93)(43/2)]2.43/1000 = 0.090$$
$$b_{15} = [(43/2)+(0.89)(56/2)]2.43/1000 = 0.109$$
$$b_{16} = (56/2)2.43/1000 = 0.067.$$

The equation $\mathbf{G}\,\mathbf{N}(t) = \mathbf{N}(t + \Delta t)$ now reads

$$\begin{pmatrix} 0 & 0 & 0.036 & 0.09 & 0.109 & 0.067 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.86 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.93 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.89 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.95 & 0 \end{pmatrix} \begin{pmatrix} 10 \\ 14 \\ 15 \\ 18 \\ 22 \\ 24 \end{pmatrix} = \begin{pmatrix} 3 \\ 10 \\ 12 \\ 14 \\ 16 \\ 21 \end{pmatrix}$$

Thus the difference between the 1945 women-population totals—as given and as computed—is only (80) – (30 + 10 + 12 + 14 + 16 + 21) = 80 – 76 = 4, or 400 women. The difference is attributable to the "truncated" first entry (or the 0–4 year group).

REFERENCES

Chan, Y.; Rasmussen, W. (1979). "Forecasting housing requirements in a college town." *Journal of the Urban Planning and Development Division* 105: 9–23.

Jha, K. (1972). Demographic models. Working Paper, Department of Civil Engineering, Pennsylvania State University, University Park, Pennsylvania.

Patterson, T. S. (1995). Dynamic maintenance scheduling for a stochastic telecommunications network: Determination of performance factors. Master's Thesis. Department of Operational Sciences. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson Air AFB, Ohio.

Patterson, K.; Horton, K. G.; Chan, Y. (1994). Games with multiple payoffs. Working Paper, Department of Operational Sciences, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio.

Wright, S.; Chan, Y. (1994). Pure and polluted ground water classification on a pixel map. Working Paper. Department of Operational Sciences, Air Force Institute of Technology, Wright-Paterson AFB, Ohio.

Wright, S.; Chan, Y. (1994a). MCDM applied to the ICM contextual image classification Technique. Working Paper. Department of Operational Sciences, Air Force Institute of Technology, Wright-Paterson AFB, Ohio.

Zelany, M. (1982). *Multiple criteria decision making.* New York: McGraw-Hill.

Index