TUTORIALS IN
OPERATIONS RESEARCH

2006

Tutorials in Operations Research

Models, Methods, and Applications for Innovative Decision Making

Michael P. Johnson, Bryan Norman, and Nicola Secomandi,
Tutorials Co-Chairs and Volume Editors

Paul Gray, Series Editor
Harvey J. Greenberg, Series Founder

Presented at the INFORMS Annual Meeting, November 5–8, 2006

informs® Institute for Operations Research and the Management Sciences

www.informs.org

ISBN 13 978-1-877640-20-9
ISBN 1-877640-20-4

To order this book, contact:

INFORMS
7240 Parkway Drive, Suite 310
Hanover, MD 21076 USA
Phone: (800) 4-INFORMS or (443) 757-3500
Fax: (443) 757-3515
E-mail: informs@informs.org
URL: www.informs.org

TUTORIALS IN
OPERATIONS RESEARCH
INFORMS 2006

ISBN13 13 978-1-877640-20-9

Table of Contents

TUTORIALS IN
OPERATIONS RESEARCH
INFORMS 2006

ISBN13 978-1-877640-20-9

Foreword

John D. C. Little has long told us that the primary role of a professional society is to disseminate knowledge. Tutorials are the lifeblood of our professional society. They help introduce people to fields about which they previously knew little. They stimulate people to examine problems they would not otherwise have considered. They help point people to the state of the art and important unsolved problems. It is no surprise that tutorials are one of the major activities at the INFORMS annual meetings.

Each year, about 15 tutorials are presented at the INFORMS meeting. Although the attendance at tutorial sessions is among the largest of all sessions—numbers around 200 are common—until two years ago, their important content was lost to the many INFORMS members who could not attend the tutorial sessions or the annual meeting itself. Clearly, INFORMS was underusing one of its treasures.

In 2003, Harvey Greenberg of the University of Colorado at Denver (founding editor of the *INFORMS Journal on Computing* and well-known for his many contributions to OR scholarship and professional service) was appointed the Tutorials Chair for the Denver meeting. He recognized the problem of a lack of institutional memory about tutorials and decided to do something. He organized the Tutorials in Operations Research series of books. His idea was that a selection of the tutorials offered at the annual meeting would be prepared as chapters in an edited volume widely available through individual and library purchase. To ensure its circulation, the book would be available at the INFORMS annual fall meeting.

Harvey edited the *TutORials* book for the Denver INFORMS meeting in 2004, which was published by Springer. In 2005, Frederick H. Murphy (then Vice President of Publications for INFORMS), working closely with Harvey, convinced the INFORMS Board of Directors to bring the annual *TutORials* volume under the umbrella of our society. Harvey was appointed Series Editor. He, in turn, asked J. Cole Smith of the University of Florida and Tutorials Chair of the San Francisco annual meeting to serve as editor of the 2005 volume, the first to be published by INFORMS. In doing so, Harvey initiated the policy that the invited Tutorials Chair also serve as the Volume Editor. As the result of a suggestion by Richard C. Larson, 2005 President of INFORMS, a CD version of the volume was also made available. In mid-2005, Harvey Greenberg asked to relinquish the series editorship. I was appointed to replace him.

This year, the Pittsburgh meeting Chair, Michael Trick, appointed three Tutorials Co-Chairs—Michael P. Johnson and Nicola Secomandi of Carnegie Mellon University, and Bryan Norman of the University of Pittsburgh—who serve as coeditors of this volume. They have assembled nine tutorials for this volume that, as in previous years, cover a broad range of fields within OR. These tutorials include the following.

- Deterministic mathematical programming
- Mathematical programming under uncertainty
- Dynamic programming
- OR practice
- Production and inventory management
- Game theory applied to supply chain interactions
- Supply chain networks

The authors are a truly diverse, international group that comes from major universities including Cornell, Eindhoven (The Netherlands), Kent (United Kingdom), Lehigh,

Massachusetts (Boston), Michigan, MIT, Northwestern, Rutgers, University of California, Berkeley, University of California, Santa Barbara, and the University of Pennsylvania's Wharton School.

On behalf of the INFORMS membership. I thank the three coeditors for their vision in creating this year's tutorial series and doing the enormous amount of work required to create this volume. INFORMS is also indebted to the authors who contributed the nine chapters.

The TutORials series also benefits from the work of its Advisory Committee, consisting of Erhan Erkut (Bilkent University, Turkey), Harvey J. Greenberg (University of Colorado at Denver and Health Sciences Center), Frederick S. Hillier (Stanford University), J. Cole Smith (University of Florida), and David Woodruff (University of California, Davis)

Finally, an important thank you to Molly O'Donnell (Senior Production Editor), Patricia Shaffer (Director of Publications), and the members of the publications staff at the INFORMS office for the physical preparation of this volume and its publication in a timely manner.

Paul Gray
Series Editor
Claremont Graduate University
Claremont, California

Acknowledgments

Our deep gratitude goes to the authors of the chapters in this volume, who worked diligently in the face of a challenging production schedule to prepare well-written and informative tutorials. Paul Gray, Series Editor, provided useful editorial guidance that streamlined our tasks. Patricia Shaffer, INFORMS Director of Publications, and Molly O'Donnell, INFORMS Senior Production Editor, gently nudged us to complete our work in time for final production. We thank Series Founder Harvey Greenberg for his work establishing the *TutORials* website and conveying valuable institutional history to guide our work. We thank Mike Trick, Chair of the INFORMS Pittsburgh 2006 organizing committee, for encouraging the three of us to arrange the cluster of invited tutorial sessions and editing this volume. Finally, we thank each other for cooperation amidst the many e-mails and phone calls that enabled us to work as efficiently as possible.

Michael P. Johnson
Bryan Norman
Nicola Secomandi

TUTORIALS IN
OPERATIONS RESEARCH
INFORMS 2006

ISBN13 978-1-877640-20-9

Preface

This volume of *Tutorials in Operations Research*, subtitled "Models, Methods, and Applications for Innovative Decision Making," is the third in a series that started with the volume edited by Harvey Greenberg and published by Springer in 2004. Like the previous volume of *TutORials* (which was edited by J. Cole Smith, published by INFORMS, and made available at the 2005 INFORMS meeting in San Francisco, CA), the present volume continues an innovative tradition in scholarship and academic service. First, all of the chapters in this volume correspond to tutorial presentations made at the 2006 INFORMS meeting held in Pittsburgh, PA. This conveys a sense of immediacy to the volume: readers have the opportunity to gain knowledge on important topics in OR/MS quickly, through presentations and the written chapters to which they correspond. Second, the chapters in this volume span the range of OR/MS sectors that make this field exciting and relevant to academics and practitioners alike: analytic methods (deterministic and dynamic math programming and math programming under risk and uncertainty), application areas (production and inventory management, interactions between supply chain actors, and supply chain network design), and OR/MS practice (spreadsheet modeling and analysis).

We believe that this volume, like its predecessors, will serve as a reference guide for best practices and cutting-edge research in OR/MS: It is a "go-to" guide for operations researchers. Moreover, the topics covered here are consistent with the theme of the current conference: a "renaissance" in operations research that has resulted in new theory, computational models, and applications that enable public and private organizations to identify new business models and develop competitive advantages.

The administrative challenges of producing a volume of tutorials to coincide with the conference at which the tutorials are presented has been significant. The three Volume Editors, who are also the Tutorials Co-Chairs of the conference presentations, are fortunate to have relied on the excellent model of last year's volume, as well as the guidance of Paul Gray, Series Editor. We now review the topics and findings of the nine chapters that comprise this volume.

Linear programming is one of the fundamental tools of operations research and has been at the core of operations research applications since the middle of the last century. Since the initial introduction of the simplex method, many ideas have been introduced to improve problem solution times. Additionally, the advent of interior point methods has provided an alternative method for solving linear programs that has drawn considerable interest over the last 20 years. In Chapter 1, "Linear Equations, Inequalities, Linear Programs, and a New Efficient Algorithm," Katta G. Murty discusses the history of linear programming, including both the simplex method and interior point methods, and discusses current and future directions in solving linear programs more efficiently.

Math programming contains a number of extensions to conventional modeling frameworks that allow the solution of otherwise intractable real-world problems. One example of this is semidefinite and second-order cone programming, examined by Farid Alizadeh in "Semidefinite and Second-Order Cone Programming and Their Application to Shape-Constrained Regression and Density Estimation." Using the fundamental definitions of positive semidefinite matrices and membership in cones and second-order cones, Alizadeh shows that semidefinite programs (SDP) and second-order cone programs (SOCP) have a num-

ber of the duality, complementarity, and optimality properties associated with conventional linear programs. In addition, there are interior point algorithms for both SDP and SOCP that enable the solution of realistically sized instances of SDP and SOCP. Alizadeh applies SOCP to parametric and nonparametric shape-constrained regression and applies a hybrid of SDP and SOCP to parametric and nonparametric density function estimation. Finally, Alidazeh describes a promising real-world application of SDP and SOCP: approximation of the arrival rate of a nonhomogenous Poisson process with limited arrivals data.

Many operations research methods are based on knowing problem data with certainty. However, in many real applications, problem data such as resource levels, cost information, and demand forecasts are not known with certainty. Many stochastic optimization methods have been developed to model problems with stochastic problem data. These methods are limited by the assumption that problem uncertainty can be characterized by a distribution with known parameters, e.g., demand follows a normal distribution with a given mean and variance. In "Model Uncertainty, Robust Optimization, and Learning" Andrew E. B. Lim, J. George Shanthikumar, and Z. J. Max Shen discuss methods that can be applied to problems where the problem uncertainty is more complex. The authors propose robust optimization approaches that can be applied to these more general problems. The methods are discussed from a theoretical perspective and are applied in inventory and portfolio selection problems.

In the next chapter, Dimitris Bertsimas and Aurélie Thiele ("Robust and Data-Driven Optimization: Modern Decision Making Under Uncertainty") consider an important aspect of decision making under uncertainty: robust optimization approaches. Many approaches to solving this problem result in very conservative policies because the policy is based on considering the worst-case scenario. Bertsimas and Thiele provide a framework that provides a more comprehensive approach that goes beyond just considering the worst-case scenario. Moreover, this approach can incorporate the decision maker's risk preferences in determining an operating policy. Bertsimas and Thiele discuss the theory underlying their methods and present applications to portfolio and inventory management problems.

Many operations research problems involve the allocation of resources over time or under conditions of uncertainty. In "Approximate Dynamic Programming for Large-Scale Resource Allocation Problems," Warren B. Powell and Huseyin Topaloglu present modeling and solution strategies for the typical large-scale resource allocation problems that arise in these contexts. Their approach involves formulating the problem as a dynamic program and replacing its value function with tractable approximations, which are obtained by using simulated trajectories of the system and iteratively improving on some initial estimates. Consequently, the original complex problem decomposes into time-staged subproblems linked by value function approximations. The authors illustrate their approach with computational experiments, which indicate that the proposed strategies yield high-quality solutions, and compare it with conventional stochastic programming methods.

Spreadsheets are ubiquitous in business and education for data management and analysis. However, there is often a tension between the need for quick analyses, which may result in errors and use of only a small fraction of a spreadsheet software's features, and the need for sophisticated understanding of the capabilities and features of spreadsheets, which may require time-intensive training. In "Enhance Your Own Research Productivity Using Spreadsheets," Janet M. Wagner and Jeffrey Keisler remind us of the high stakes of many "mission-critical" spreadsheet-based applications and the significant likelihood of errors in these applications. In response to these identified needs, Wagner and Keisler argue for the importance of spreadsheet-based methods and tools for data analysis, user interface design, statistical modeling, and math programming that may be new even to experienced users. The authors' presentation of important features of Microsoft Excel relevant to OR/MS researchers and practitioners is framed by four case studies drawn from education and business and available online.

The theory on multiechelon production/inventory systems lies at the core of supply chain management. It provides fundamental insights that can be used to design and manage supply chains, both at the tactical and operational planning levels. In "Multiechelon Production Inventory Systems: Optimal Policies, Heuristics, and Algorithms," Geert-Jan van Houtum presents the main concepts underlying this theory. He illustrates those systems for which the structure of the optimal policy is known, emphasizing those features of the system that are necessary to obtain such a structure, and discusses appropriate heuristic methods for those systems for which the structure of the optimal policy is unknown. Special attention is given to describing the class of basestock policies and conditions that make such policies, or generalizations thereof, optimal.

While tactical and operational considerations are clearly important in managing a supply chain, recent years have witnessed increased attention by operations management researchers to applying game-theoretic concepts to analyze strategic interactions among different players along a supply chain. The next chapter, written by Gérard P. Cachon and Serguei Netessine ("Game Theory in Supply Chain Analysis"), provides a detailed survey of this literature. Cachon and Netessine illustrate the main game-theoretic concepts that have been applied, but also point out those concepts that have potential for future applications. In particular, they carefully discuss techniques that can be used to establish the existence and uniqueness of equilibrium in noncooperative games. The authors employ a newsvendor game throughout the chapter to illustrate the main results of their analysis.

Many important extensions to basic models of supply chain management address demand uncertainty—the possibility that fluctuations in demand for goods provided by a supply chain could result in service disruptions. In "Planning for Disruptions in Supply Chain Networks," Lawrence V. Snyder, Maria P. Scaparra, Mark S. Daskin, and Richard L. Church develop planning models that address uncertainty in the supply of goods and services arising from disruptions that might close product facilities. Their key insight is that models accounting for demand uncertainty use results in risk pooling effects to argue for fewer distribution centers, while those that account for supply uncertainty generally result in more distribution facilities to preserve the robustness of the network. The authors present models that address the location of facilities alone versus the construction of entire distribution networks, distinguish between supply chain design de novo and fortification of existing systems, and address uncertainty through minimizing worst-case outcomes, expected cost, and maximum regret.

We hope that you find this collection of tutorials stimulating and useful. *TutORials* represents the best that INFORMS has to offer: theory, applications, and practice that are grounded in problems faced by real-world organizations, fortified by advanced analytical methods, enriched by multidisciplinary perspectives, and useful to end-users, be they teachers, researchers, or practitioners.

Michael P. Johnson
Carnegie Mellon University
Pittsburgh, Pennsylvania

Bryan Norman
University of Pittsburgh
Pittsburgh, Pennsylvania

Nicola Secomandi
Carnegie Mellon University
Pittsburgh, Pennsylvania

TUTORIALS IN
OPERATIONS RESEARCH
INFORMS 2006

 | ISBN 13 978-1-877640-20-9
DOI 10.1287/educ.1063.0024

Linear Equations, Inequalities, Linear Programs, and a New Efficient Algorithm

Katta G. Murty
Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, Michigan 48109-2117, murty@umich.edu

Abstract The dawn of mathematical modeling and algebra occurred well over 3,000 years ago in several countries (Babylonia, China, India, etc.). The earliest algebraic systems constructed are systems of linear equations, and soon after the famous elimination method for solving them was discovered in China and India. This effort culminated in the writing of two books that attracted international attention by the Arabic mathematician Muhammad ibn-Musa Alkhawarizmi in the first half of the nineth century. The first, *Al-Maqala fi Hisab al-jabr w'almuqabilah* (*An Essay on Algebra and Equations*), was translated into Latin under the title *Ludus Algebrae*; the name "algebra" for the subject came from this Latin title, and Alkhawarizmi is regarded as the father of algebra. Linear algebra is the branch of algebra dealing with systems of linear equations. The second book, *Kitab al-Jam'a wal-Tafreeq bil Hisab al-Hindi*, appeared in Latin translation under the title *Algoritmi de Numero Indorum* (meaning *Alkhawarizmi Concerning the Hindu Art of Reckoning*), and the word "algorithm" (meaning procedures for solving algebraic systems) originated from this Latin title.

The elimination method for solving linear equations remained unknown in Europe until Gauss rediscovered it in the nineteenth century while approximating by a quadratic formula the orbit of the asteroid Ceres based on recorded observations in tracking it earlier by the Italian astronomer Piazzi. Europeans gave the names "Gaussian elimination method," "GJ (Gauss-Jordan) elimination method" for this method.

However, until recently, there was no computationally viable method to solve systems of linear constraints including inequalities. Examples of linear constraints with inequalities started appearing in published literature in the mid-eighteenth century. In the nineteenth and early twentieth centuries, Fourier, De la Vallée Poussin, Farkas, Kantarovich, and others did initial work for solving such systems. This work culminated in the 1947 paper on the simplex method for linear programming (LP) by George Dantzig. The simplex method is a one-dimensional boundary method; it quickly became the leading algorithm to solve LPs and related problems. Its computational success made LP a highly popular modeling tool for decision-making problems, with numerous applications in all areas of science, engineering, and business management. From the nature of the simplex method, LP can be viewed as the twentieth-century extension of linear algebra to handle systems of linear constraints including inequalities.

Competing now with the simplex method are a variety of interior point methods for LP, developed in the last 20 years and stimulated by the pioneering work of Karmarkar; these follow a central path using a logarithmically defined centering strategy. All these methods and also the simplex method need matrix inversions; their success for large-scale problem solving requires taking careful advantage of sparcity in the data.

I will discuss a new interior point method based on a much-simpler centering strategy that I developed recently. It is a fast, efficient descent method that can solve LPs without matrix inversions; hence, it can handle dense problems and is also not affected by redundant constraints in the model.

Keywords linear programming; Dantzig's simplex method; boundary methods; gravitational methods; interior point methods; solving LPs without matrix inversions

1. Historical Overview

1.1. Mathematical Modeling, Algebra, Systems of Linear Equations, and Linear Algebra

One of the most fundamental ideas of the human mind, discovered more than 5,000 years ago by the Chinese, Indians, Iranians, and Babylonians, is to represent the quantities that we like to determine by symbols; usually letters of the alphabet such as x, y, z; and then express the relationships between the quantities represented by these symbols in the form of equations, and finally use these equations as tools to find out the true values represented by the symbols. The symbols representing the unknown quantities to be determined are nowadays called *unknowns* or *variables* or *decision variables*.

The process of representing the relationships between the variables through equations or other functional relationships is called *modeling* or *mathematical modeling*. The earliest mathematical models constructed are systems of linear equations, and soon after the famous *elimination method* for solving them was discovered in China and India. The Chinese text *Chiu-Chang Suanshu* (*9 Chapters on the Mathematical Art*), composed over 2,000 years ago, describes the method using a problem of determining the yield (measured in units called "tou") from three types of grain: inferior, medium, superior; given the yield data from three experiments each using a separate combination of the three types of grain (see Kangshen et al. [14] for information on this ancient work; also a summary of this ancient Chinese text can be seen at the website: http://www-groups.dcs.st-and.ac.uk/~history/HistTopics/Nine_chapters.html). Ancient Indian texts, *Sulabha suutrah* (*Easy Solution Procedures*) with origins to the same period, describe the method in terms of solving systems of two linear equations in two variables (see Lakshmikantham and Leela [18] for information on these texts, and for a summary and review of this book see http://www.tlca.com/adults/origin-math.html).

This effort culminated around 825 AD in the writing of two books by the Arabic mathematician Muhammad ibn-Musa Alkhawarizmi that attracted international attention. The first was *Al-Maqala fi Hisab al-jabr w'almuqabilah* (*An Essay on Algebra and Equations*). The term "al-jabr" in Arabic means "restoring" in the sense of solving an equation. In Latin translation, the title of this book became *Ludus Algebrae*, the second word in this title surviving as the modern word *algebra* for the subject, and Alkhawarizmi is regarded as the father of algebra. *Linear algebra* is the name given subsequently to the branch of algebra dealing with systems of linear equations. The word *linear* in "linear algebra" refers to the "linear combinations" in the spaces studied, and the *linearity* of "linear functions" and "linear equations" studied in the subject.

The second book, *Kitab al-Jam'a wal-Tafreeq bil Hisab al-Hindi*, appeared in a Latin translation under the title *Algoritmi de Numero Indorum*, meaning *Al-Khwarizmi Concerning the Hindu Art of Reckoning*; it was based on earlier Indian and Arabic treatises. This book survives only in its Latin translation, because all copies of the original Arabic version have been lost or destroyed. The word *algorithm* (meaning procedures for solving algebraic systems) originated from the title of this Latin translation. Algorithms seem to have originated in the work of ancient Indian mathematicians on rules for solving linear and quadratic equations.

1.2. Elimination Method for Solving Linear Equations

We begin with an example application that leads to a model involving simultaneous linear equations. A steel company has four different types of scrap metal (called SM-1 to SM-4) with compositions given in Table 1 below. They need to blend these four scrap metals into a mixture for which the composition by weight is Al-4.43%, Si-3.22%, C-3.89%, Fe-88.46%. How should they prepare this mixture?

Table 1. Compositions of available scrap metals.

	% in type, by weight, of element			
Type	Al	Si	C	Fe
SM-1	5	3	4	88
SM-2	7	6	5	82
SM-3	2	1	3	94
SM-4	1	2	1	96

To answer this question, we first define the decision variables, denoted by x_1, x_2, x_3, x_4, where for $j = 1$ to 4, $x_j =$ proportion of SM-j by weight in the mixture to be prepared. Then the percentage by weight of the element Al in the mixture will be $5x_1 + 7x_2 + 2x_3 + x_4$, which is required to be 4.43. Arguing the same way for the elements Si, C, and Fe, we find that the decision variables x_1 to x_4 must satisfy each equation in the following system of linear equations to lead to the desired mixture:

$$5x_1 + 7x_2 + 2x_3 + x_4 = 4.43$$
$$3x_1 + 6x_2 + x_3 + 2x_4 = 3.22$$
$$4x_1 + 5x_2 + 3x_3 + x_4 = 3.89$$
$$88x_1 + 82x_2 + 94x_3 + 96x_4 = 88.46$$
$$x_1 + x_2 + x_3 + x_4 = 1.$$

The last equation in the system shows that the sum of the proportions of various ingradients in a blend must always equal 1. From the definition of the variables given above, it is clear that a solution to this system of equations makes sense for the blending application under consideration only if all variables in the system have nonnegative values in it. The nonnegativity restrictions on the variables are *linear inequality constraints*. They cannot be expressed in the form of linear equations, and because nobody knew how to handle linear inequalities at that time, they ignored them and considered this system of equations as the mathematical model for the problem.

To solve a system of linear equations, each step in the elimination method uses one equation to express one variable in terms of the others, then uses that expression to eliminate that variable and that equation from the system leading to a smaller system. The same process is repeated on the remaining system. The work in each step is organized conveniently through what is now called the *Gauss-Jordan* (*GJ*) *pivot step*. We will illustrate this step on the following system of three linear equations in three decision variables given in the following detached coefficient tableau (Table 2, top). In this representation, each row in the tableau corresponds to an equation in the system, and RHS is the column vector of right side constants in the various equations. Normally, the equality symbol for the equations is omitted.

Table 2. An illustration of the GJ pivot step.

Basic variable	x_1	x_2	x_3	RHS
	$\boxed{1}$	−1	−1	10
	−1	2	−2	20
	1	−2	−4	30
x_1	1	−1	−1	10
	0	1	−3	30
	0	−1	−3	20

In this step on the system given in the top tableau, we are eliminating the variable x_1 from the system using the equation corresponding to the first row. The column vector of the variable eliminated, x_1, is called the *pivot column*, and the row of the equation used to eliminate the variable is called the *pivot row* for the pivot step, the element in the pivot row and pivot column, known as the *pivot element*, is boxed. The pivot step converts the pivot column into the unit column with "1" entry in the pivot row and "0" entries in all other rows. In the resulting tableau after this pivot step is carried out, the variable eliminated, x_1, is recorded as the *basic variable* in the pivot row. This row now contains an expression for x_1 as a function of the remaining variables. The other rows contain the remaining system after x_1 is eliminated; the same process is now repeated on this system.

When the method is continued on the remaining system, two things may occur: (a) all entries in a row may become 0, this is an indication that the constraint in the corresponding row in the original system is a redundant constraint, such rows are eliminated from the tableau; and (b) the coefficients of all the variables in a row may become 0, while the RHS constant remains nonzero, this indicates that the original system of equations is inconsistent, i.e., it has no solution, if this occurs, the method terminates.

If the inconsistency termination does not occur, the method terminates after performing pivot steps in all rows. If there are no nonbasic variables at that stage, equating each basic variable to the RHS in the final tableau gives the unique solution of the system. If there are nonbasic variables, from the rows of the final tableau, we get the general solution of the system in parametric form in terms of the nonbasic variables as parameters.

The elimination method remained unknown in Europe until Gauss rediscovered it at the beginning of the nineteenth century while calculating the orbit of the asteroid Ceres based on recorded observations in tracking it earlier. It was lost from view when Piazzi, the astronomer tracking it, fell ill. Gauss got the data from Piazzi, and tried to approximate the orbit of Ceres by a quadratic formula using that data. He designed the method of least squares for estimating the best values for the parameters to give the closest fit to the observed data; this gives rise to a system of linear equations to be solved. He rediscovered the elimination method to solve that system. Even though the system was quite large for hand computation, Gauss's accurate computations helped in relocating the asteroid in the skies in a few months time, and his reputation as a mathematician soared.

Europeans gave the names *Gaussian elimination method*, *Gauss-Jordan elimination method* to two variants of the method at that time. These methods are still the leading methods in use today for solving systems of linear equations.

1.3. Lack of a Method to Solve Linear Inequalities Until Modern Times

Even though linear equations had been conquered thousands of years ago, systems of linear inequalities remained inaccessible until modern times. The set of feasible solutions to a system of linear inequalities is called a *polyhedron* or *convex polyhedron*, and geometric properties of polyhedra were studied by the Egyptians earlier than 2000 BC while building the pyramids, and later by the Greeks, Chinese, Indians, and others.

The following theorem (for a proof see Monteiro and Adler [24]) relates systems of linear inequalities to systems of linear equations.

Theorem 1. *If the system of linear inequalities:* $A_{i.}x \geq b_i$, $i = 1$ *to* m *in variables* $x = (x_1, \ldots, x_n)^T$ *has a feasible solution, then there exists a subset* $\mathbf{P} = \{p_1, \ldots, p_s\} \subset \{1, \ldots, m\}$ *such that every solution of the system of linear equations:* $A_{i.}x = b_i, i \in \mathbf{P}$ *is also feasible to the orignal system of linear inequalities.*

A paradox: Theorem 1 presents an interesting paradox. As you know, linear equations can be transformed into linear inequalities by replacing each equation with the opposing pair of inequalities. However, there is no way a linear inequality can be transformed into

linear equations. This indicates that linear inequalities are more fundamental than linear equations.

This theorem shows, however, that linear equations are the key to solving linear inequalities, and hence are more fundamental.

Theorem 1 provides an enumerative approach for solving a system of linear inequalities, involving enumeration over subsets of the inequalities treated as equations. But the effort required by the method grows exponentially with the number of inequalities in the system in the worst case.

1.4. The Importance of Linear Inequality Constraints and Their Relation to Linear Programs

The first interest in inequalities arose from studies in mechanics, beginning with the eighteenth century.

Linear programming (*LP*) involves optimization of a linear objective function subject to linear inequality constraints. Crude examples of LP models started appearing in published literature from about the mid-eighteenth century. We will now present an example of a simple application of LP from the class of *product mix models* from Murty [26, 31].

A fertilizer company makes two kinds of fertilizers called hi-phosphate (Hi-ph) and lo-phosphate (Lo-ph). The manufacture of these fertilizers requires three raw materials called RM 1, RM 2, RM 3. At present, their supply of these raw materials comes from the company's own quarry, which can only supply maximum amounts of 1,500, 1,200, and 500 tons/day, respectively, of RM 1, RM 2, and RM 3. Although other vendors can supply these raw materials if necessary, at the moment, the company is not using these outside suppliers.

The company sells its output of Hi-ph and Lo-ph fertilizers to a wholesaler willing to buy any amount the company can produce, so there are no upper bounds on the amounts of Hi-ph and Lo-ph manufactured daily.

At the present rates of operation, cost accounting department estimates that it is costing the quarry \$50, \$40, \$60/ton, respectively, to produce and deliver RM 1, RM 2, RM 3 at the fertilizer plant. Also, at the present rates of operation, all other production costs (for labor, power, water, maintenance, depreciation of plant and equipment, floor space, insurance, shipping to the wholesaler, etc.) come to \$7/ton to manufacture Hi-ph or Lo-ph and to deliver them to the wholesaler.

The sale price of the manufactured fertilizers to the wholesaler fluctuates daily, but averages over the last one month have been \$222, \$107/ton, respectively, for Hi-Ph and Lo-ph fertilizers.

The Hi-ph manufacturing process needs as inputs two tons of RM 1, and one ton each of RM 2, RM 3 for each ton of Hi-ph manufactured. Similarly, the Lo-ph manufacturing process needs as inputs one ton of RM 1, and one ton of RM 2 for each ton of Lo-ph manufactured. So, the net profit/ton of fertilizer manufactured is $\$(222-2\times 50-1\times 40-1\times 60-7)=15$, $(107-1\times 50-1\times 40-7)=10$/respectively, for Hi-ph, Lo-ph.

We will model the problem with the aim of determining how much of Hi-ph and Lo-ph to make daily to maximize the total daily net profit from these fertilizer operations. Clearly, two decision variables exist; these are

$$x_1 = \text{the tons of Hi-ph made per day}$$
$$x_2 = \text{the tons of Lo-ph made per day.}$$

Because all data is given on a per ton basis, this indicates that the linearity assumptions (proportionality, additivity) are quite reasonable in this problem to express each constraint and the objective function. Also, the amount of each fertilizer manufactured can vary continuously within its present range. So, LP is an appropriate model for this problem. The LP

formulation of this fertilizer product mix problem is given below. Each constraint in the model is the material balance inequality of the *item* shown against it.

$$\begin{array}{lrl} \text{Maximize} & z(x) = 15x_1 + 10x_2 & \text{Item} \\ \text{subject to} & 2x_1 + x_2 \leq 1500 & \text{RM 1} \\ & x_1 + x_2 \leq 1200 & \text{RM 2} \\ & x_1 \leq 500 & \text{RM 3} \\ & x_1 \geq 0, \quad x_2 \geq 0 & \end{array} \tag{1}$$

In this example, all constraints on the variables are inequality constraints. In the same way, inequality constraints appear much more frequently and prominently than equality constraints in most real-world applications. In fact, we can go as far as to assert that in most applications in which a linear model is the appropriate one to use, most constraints are actually linear inequalities, and linear equations play only the role of a computational tool through approximations, or through results similar to Theorem 1. Linear equations were used to model problems mostly because an efficient method to solve them is known.

Fourier was one of the first to recognize the importance of inequalities as opposed to equations for applying mathematics. Also, he is a pioneer who observed the link between linear inequalities and linear programs, in the early nineteenth century.

For example, the problem of finding a feasible solution to the following system of linear inequalities (2) in x_1, x_2, can be posed as another LP for which an initial feasible solution is readily available. Formulating this problem, known as a *Phase I problem*, introduces one or more nonnegative variables known as *artificial variables* into the model. All successful LP algorithms require an initial feasible solution, so the Phase I problem can be solved using any of those algorithms, and at termination, it either outputs a feasible solution of the original problem, or an evidence for its infeasibility. The Phase I model for finding a feasible solution for (2) is (3), it uses one artificial variable x_3.

$$\begin{aligned} x_1 + 2x_2 &\geq 10 \\ 2x_1 - 4x_2 &\geq 15 \\ -x_1 + 10x_2 &\geq 25 \end{aligned} \tag{2}$$

$$\begin{array}{lr} \text{Minimize} & x_3 \\ \text{subject to} & x_1 + 2x_2 + x_3 \geq 10 \\ & 2x_1 - 4x_2 + x_3 \geq 15 \\ & -x_1 + 10x_2 + x_3 \geq 25 \\ & x_3 \geq 0 \end{array} \tag{3}$$

For the Phase I problem (3), $(x_1, x_2, x_3)^T = (0, 0, 26)^T$ is a feasible solution. In fact, solving such a Phase I problem provides the most efficient approach for solving systems of linear inequalities.

Also, the duality theory of linear programming shows that any linear program can be posed as a problem of solving a system of linear inequalities without any optimization. Thus, solving linear inequalities, and LPs, are mathematically equivalent problems. Both problems of comparable sizes can be solved with comparable efficiencies by available algorithms. So, the additional aspect of "optimization" in linear programs does not make LPs any harder either theoretically or computationally.

1.5. Elimination Method of Fourier for Linear Inequalities

By 1827, Fourier generalized the elimination method to solve a system of linear inequalities. The method, now known as the *Fourier* or *Fourier-Motzkin elimination method*, is one of the earliest methods proposed for solving systems of linear inequalities. It consists of successive elimination of variables from the system. We will illustrate one step in this method using an example in which we will eliminate the variable x_1 from the following system.

$$\begin{aligned} x_1 - 2x_2 + x_3 &\le 6 \\ 2x_1 + 6x_2 - 8x_3 &\le -6 \\ -x_1 - x_2 - 2x_3 &\le 2 \\ -2x_1 - 6x_2 + 2x_3 &\le 2 \end{aligned}$$

x_1 appears with a positive coefficient in the first and second constraints, and a negative coefficient in the third and fourth constraints. By making the coefficient of x_1 in each constraint into 1, these constraints can be expressed as

$$\begin{aligned} x_1 &\le 6 + 2x_2 - x_3 \\ x_1 &\le -3 - 3x_2 + 4x_3 \\ -2 - x_2 - 2x_3 &\le x_1 \\ -1 - 3x_2 + x_3 &\le x_1. \end{aligned}$$

The remaining system after x_1 is eliminated is therefore

$$\begin{aligned} -2 - x_2 - 2x_3 &\le 6 + 2x_2 - x_3 \\ -2 - x_2 - 2x_3 &\le -3 - 3x_2 + 4x_3 \\ -1 - 3x_2 + x_3 &\le 6 + 2x_2 - x_3 \\ -1 - 3x_2 + x_3 &\le -3 - 3x_2 + 4x_3 \end{aligned}$$

and then $\max\{-2 - x_2 - 2x_3, -1 - 3x_2 + x_3\} \le x_1 \le \min\{6 + 2x_2 - x_3, -3 - 3x_2 + 4x_3\}$ is used to get a value for x_1 in a feasible solution when values for other variables are obtained by applying the same steps on the remaining problem successively.

However, starting with a system of m inequalities, the number of inequalities can jump to $O(m^2)$ after eliminating only one variable from the system, thus, this method is not practically viable except for very small problems.

1.6. History of the Simplex Method for LP

In 1827, Fourier published a geometric version of the principle behind the simplex algorithm for a linear program (vertex to vertex descent along the edges to an optimum, a rudimentary version of the simplex method) in the context of a specific LP in three variables (an LP model for a Chebyshev approximation problem), but did not discuss how this descent can be accomplished computationally on systems stated algebraically. In 1910, De la Vallée Poussin designed a method for the Chebyshev approximation problem that is an algebraic and computational analogue of this Fourier's geometric version; this procedure is essentially the primal simplex method applied to that problem.

In a parallel effort, Gordan [11], Farkas [9], and Minkowski [22] studied linear inequalities, and laid the foundations for the algebraic theory of polyhedra, and derived necessary and sufficient conditions for a system of linear constraints, including linear inequalities, to have a feasible solution.

Studying LP models for organizing and planning production, Kantorovich [15] developed ideas of dual variables ("resolving multipliers") and derived a dual-simplex type method

for solving a general LP. Full citations for references before 1939 mentioned so far can be seen from the list of references in Dantzig [5] or Schrijver [37].

This work culminated in the mid-twentieth century with the development of the primal simplex method by Dantzig. This was the first complete, practically and computationally viable method for solving systems of linear inequalities. So, LP can be considered as the branch of mathematics that is an extension of linear algebra to solve systems of linear inequalities. The development of LP is a landmark event in the history of mathematics, and its application brought our ability to solve general systems of linear constraints (including linear equations, inequalities) to a state of completion.

Linear Algebra

Study of linear equations.
Originated over 2,000 years ago.

→

Linear Programming

Study of linear constraints including inequalities.
Twentieth-century extension of linear algebra.

2. The Importance of LP

LP has now become a dominant subject in the development of efficient computational algorithms, study of convex polyhedra, and algorithms for decision making. But for a short time in the beginning, its potential was not well recognized. Dantzig tells the story of how when he gave his first talk on LP and his simplex method for solving it at a professional conference, Hotelling (a burly person who liked to swim in the sea, the popular story about him was that when he does, the level of the ocean rises perceptibly (see Figures 1 and 2); my thanks to Katta Sriramamurthy for these figures) dismissed it as unimportant because everything in the world is nonlinear. But Von Neumann came to the defense of Dantzig, saying that the subject would become very important. (For an account of Von Neumann's comments at this conference, see p. xxvii of Dantzig and Thapa [6].) The preface in this book contains an excellent account of the early history of LP from the inventor of the most successful method in OR and in the mathematical theory of polyhedra.

Von Neumann's early assessment of the importance of LP (Von Neumann [39]) turned out to be astonishingly correct. Today, the applications of LP in almost all areas of science

FIGURE 1. Hotelling (a whale of a man) getting ready to swim in the ocean.

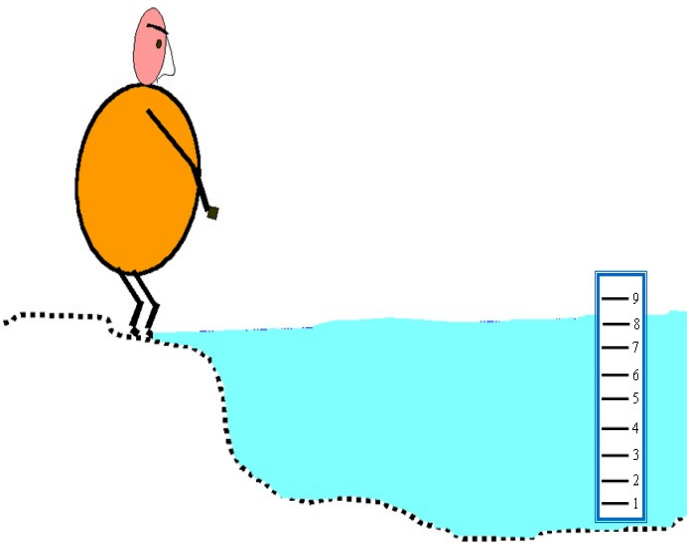
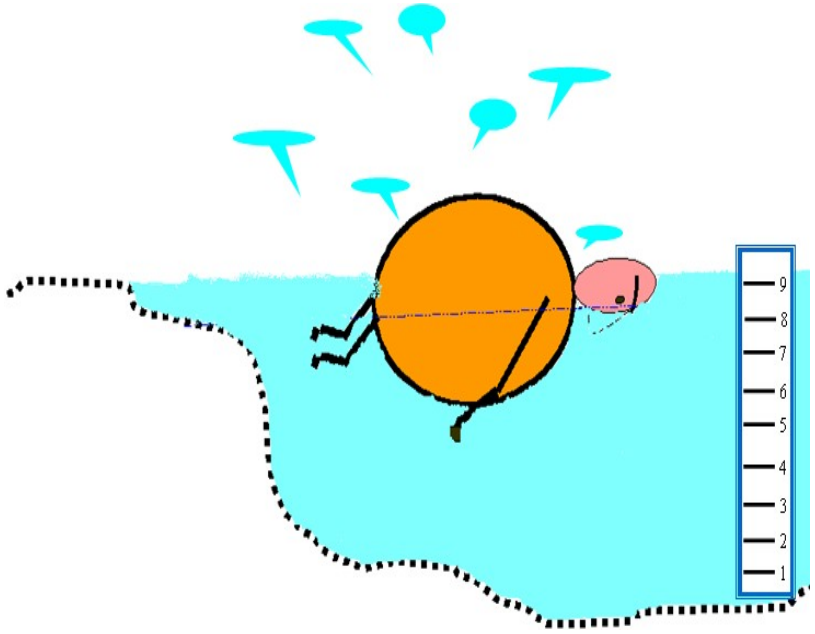


FIGURE 2. Hotelling swimming in the ocean. Watch the level of the ocean go up.



are so numerous, so well known, and recognized, that they need no enumeration. Also, LP seems to be the basis for most efficient algorithms for many problems in other areas of mathematical programming. Many successful approaches in nonlinear programming, discrete optimization, and other branches of optimization are based on LP in their iterations. Also, with the development of duality theory and game theory (Gale [10]) LP has also assumed a central position in economics.

3. Dantzig's Contributions to Linear Algebra, Convex Polyhedra, OR, and Computer Science

Much has been written about Dantzig's contributions. Also, a personal assessment of his own contributions appear in Chapter 1 of his book (Dantzig [5]). As someone who started learning LP from his course at Berkeley, I will summarize here some of his contributions usually overlooked in other statements (for a brief account of my experiences with Dantzig see Murty [32]).

3.1. Contributions to OR

The simplex method is the first effective computational algorithm for one of the most versatile mathematical models in OR. Even though LP and the simplex method for solving it originated much earlier than Dantzig's work as explained in §1.6, it started becoming prominent only with Dantzig's work, and OR was just beginning to develop around that time. The success of the simplex method is one of the root causes for the phenominal development and maturing of LP, mathematical programming in general, and OR, in the second half of the twentieth century.

3.2. Contributions to Linear Algebra and Computer Science

3.2.1. Recognizing the Irrelevance of the "RREF" Concept Emphasized in Mathematics Books on Linear Algebra. Dantzig contributed important pedagogic improvements to the teaching of linear algebra. He would state all the algorithmic steps in

the GJ elimination method using the fundamental tool of row operations on the detached coefficient tableau for the system with the variable corresponding to each column entered in a top row in every tableau. This makes it easier for young students to see that the essence of this method is to take linear combinations of equations in the original system to get an equivalent but simpler system from which a solution can be read out. In most mathematics books on linear algebra, the variables are usually left out in descriptions of the GJ method.

Also, these books state the termination condition in the GJ elimination method to be that of reaching the RREF (reduced row echelon form; a tableau is defined to be in RREF if it contains a full set of unit vectors in proper order at the left end). Dantzig (and of course a lot of other OR people) realized that it is not important that all unit vectors be at the left end of the tableau (they can be anywhere and can be scattered all over); also, it is not important that they be in proper order from left to right. He developed the very simple *data structure* (this phrase means a strategy for storing information generated during the algorithm and using it to improve the efficiency of that algorithm; perhaps this is the first instance of such a structure in computational algorithms) of associating the variable corresponding to the rth unit vector in the final tableau as the rth basic variable (or basic variable in the rth row) and storing these basic variables in a column on the tableau as the algorithm progresses. This data structure makes it easier to read the solution directly from the final tableau of the GJ elimination method by making all nonbasic variables $= 0$, and the rth basic variable $=$ the rth updated RHS constant for all r. Dantzig called this final tableau the *canonical tableau* to distinguish it from the mathematical concept of RREF. It also opened the possibility of pivot column-selection strategies instead of always selecting the leftmost eligible column in this method.

Even today, in courses on linear algebra in mathematics departments, it is unfortunate that the RREF is emphasized as the output of the GJ elimination method. For a more realistic statement of the GJ method from an OR perspective, see Murty [29].

3.2.2. Evidence (or Certificate) of Infeasibility. A fundamental theorem of linear algebra asserts that a system of linear equations is infeasible if there is a linear combination of equations in the system that is the *fundamental inconsistent equation* "$0 = a$" (where a is some nonzero number). Mathematically, in matrix notation, the statement of this theorem is: "Either the system $Ax = b$ has a solution (column) vector x, or there exists a row vector π satisfying $\pi A = 0, \pi b \neq 0$." The coefficient vector π in this linear combination is called an *evidence* (or *certificate*) of *infeasibility* for the original system $Ax = b$.

But with the usual descriptions of the GJ elimination method to get an RREF or canonical tableau, this evidence is not available when the infeasibility conclusion is reached. An important contribution of Dantzig, the *revised simplex method*, has very important consequences to the GJ elimination method. When the GJ elimination method is executed in the revised simplex format, pivot computations are not performed on the original system (it remains unchanged throughout the algorithm), but only carried out on an auxiliary matrix set up to accumulate the basis inverse, and all the computations in the algorithm are carried out using this auxiliary matrix and the data from the original system. We will call this auxiliary matrix the *memory matrix*. For solving $Ax = b$ where A is of order $m \times n$, the initial memory matrix is the unit matrix of order m set up by the side of the original system. For details of this implementation of the GJ elimination method, see §4.11 in Murty [30].

We will illustrate this with a numerical example. At the top of Table 3 is the original system in detached coefficient form on the right and the memory matrix on the left. At the bottom, we show the final tableau (we show the canonical tableau on the right just for illustration; it will not actually be computed in this implementation). BV = basic variable selected for the row; MM = memory matrix.

The third row in the final tableau represents the inconsistent equation "$0 = 2$," which shows that the original system is infeasible. The row vector of the memory matrix in this

TABLE 3. An example of an infeasible system.

				Original system				
BV	MM			x_1	x_2	x_3	x_4	RHS
	1	0	0	1	−1	1	−1	5
	0	1	0	−1	2	2	−2	10
	0	0	1	0	1	3	−3	17
				Canonical tableau				
x_1	2	1	0	1	0	4	−4	20
x_2	1	1	0	0	1	3	−3	15
	−1	−1	1	0	0	0	0	2

row, $(1, 1, -1)$, is the coefficient vector for the linear combination of equations in the original system that produces this inconsistent equation, it is the certificate of infeasibility for this system.

3.2.3. Contributions to the Mathematical Study of Convex Polyhedra. Dantzig has made fundamental contributions to the mathematical study of convex polyhedra (a classical subject being investigated by mathematicians for more than 2,000 years) when he introduced the complete version of the primal simplex method as a computational tool.

We could only see drawings of two-dimensional polyhedra before this work. Polyhedra in higher dimensions could only be visualized through imagination. The primal simplex pivot step is the first computational step for actually tracing an edge (either bounded or unbounded) of a convex polyhedron. It opened a revolutionary new computational dimension in the mathematical study of convex polyhedra, and made it possible to visualize and explore higher-dimensional polyhedra through computation. At a time when research on convex polyhedra was beginning to stagnate, the simplex method has reignited the spark, and enriched this mathematical study manyfold.

4. Algorithms Used for Solving LPs Today

Now we will summarize the main ideas behind algorithms used for solving LPs today.

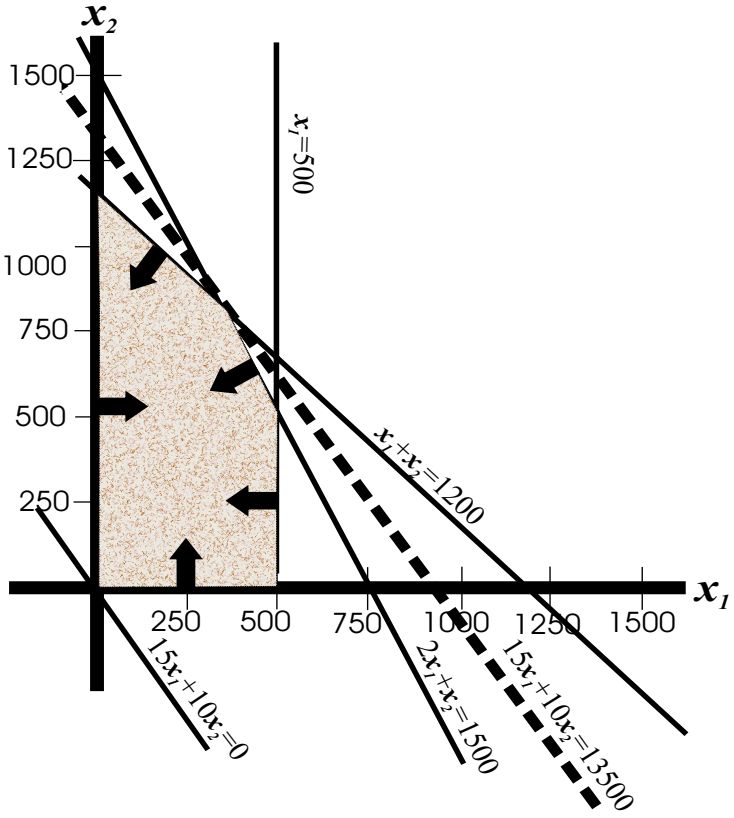
4.1. Objective Plane Sliding Geometric Method for Two-Variable LPs

This simple visual geometric method is useful for solving LPs involving only two variables by hand. Let $z(x)$ be the linear objective function we are trying to optimize. First, the feasible region is drawn on paper by hand, and then a feasible solution $\bar{x}$ is identified in it visually. Then, the objective plane (a straight line in R^2) through $\bar{x}$ represented by $z(x) = z(\bar{x})$ is drawn. Changing the RHS constant in the equation for this line (i.e., changing the objective value) is equivalent to moving this straight line parallel to itself. This objective straight line is moved parallelly in the desired direction until it reaches a stage where it is still intersecting the feasible region, but any further move in the desired direction will make it lose contact with the feasible region. The intersection of the objective straight line in this final position with the feasible region is the set of optimum solutions of the problem.

In the fertilizer product mix problem (1) from §1.4, we start with the feasible point $\bar{x} = (0, 0)$ with an objective value z_0 of 0. As z_0 is increased from 0, the line $15x_1 + 10x_2 = z_0$ moves up, keeping a nonempty intersection with the feasible region, until the line coincides with the dashed line $15x_1 + 10x_2 = 13,500$ in Figure 3 passing through the point of intersection of the two lines:

$$\begin{aligned} 2x_1 + x_2 &= 1{,}500 \\ x_1 + x_2 &= 1{,}200, \end{aligned}$$

FIGURE 3. Solution of the fertilizer product mix problem by the geometric method.



which is $\hat{x} = (300, 900)$. For any value of $z_0 > 13{,}500$, the line $15x_1 + 10x_2 = z_0$ does not intersect the feasible region. Hence, the optimum objective value in this problem is \$13,500, and the optimum solution of the problem is $\hat{x} = (300, 900)$. Hence, the fertilizer maker achieves his maximum daily net profit of \$13,500 by manufacturing 300 tons of Hi-ph and 900 tons of Lo-ph daily.

We cannot draw feasible regions for higher-dimensional LPs, so we cannot select an initial feasible solution for them visually (this itself requires solving another LP, a Phase I problem), and we cannot visually check whether the objective plane can be moved further in the desired direction without losing contact with the feasible region. Because this geometric method requires such a high degree of visibility, it has not been generalized yet to solving LPs of higher dimensions. We will show later that the new algorithm discussed in §6 is a generalization of this geometric method to higher dimensions made possible computationally through the centering step in it.

4.2. The Simplex Family of Methods (One-Dimensional Boundary Methods)

The simplex method is still the dominant algorithm in use for solving LPs. It exhibits exponential growth in the worst case, but its performance in practice has been outstanding, and is being improved continually by developments in implementation technologies. There are many variants of the simplex method, the most prominent being the primal simplex method. This method needs an initial feasible basic vector for the primal. If a primal feasible basic vector is not available, the method introduces artificial variables into the problem and sets up a Phase I problem with a readily available feasible basic vector consisting of artificial

Table 4. Original tableau.

BV	x_1	x_2	s_1	s_2	s_3	$-z$	RHS	Ratio	
s_1	2	1	1	0	0	0	1,500	1,500/2	
s_2	1	1	0	1	0	0	1,200	1,200/1	
s_3	$\boxed{1}$	0	0	0	1	0	500	500/1	PR
$-z$	15	10	0	0	0	1	0	Min = 500	

Note. All variables ≥ 0, maximize z.

basic variables. When this Phase I problem is solved by the same algorithm, at termination, it either provides a feasible basic vector for the original primal or a proof that it is infeasible.

Initiated with a feasible basic vector for the problem, the method goes through a series of GJ pivot steps exchanging one nonbasic variable for a basic variable in each (this type of basic vector change by one variable is the common feature of all variants of the simplex method). In each nondegenerate pivot step, the method moves along an edge (a *one-dimensional boundary face or corner*) of the feasible region from one basic feasible solution to an adjacent one, and the objective value strictly improves. We will illustrate with a pivot step carried out for solving the fertilizer problem (1). To solve this problem by the primal simplex method, the constraints are converted into equations by introducing slack variables s_1, s_2, s_3. The original tableau is shown in Table 4; it is also the canonical tableau with respect to the basic vector (s_1, s_2, s_3). BV = basic variable selected in the row; PC = pivot column, PR = pivot row.

The initial basic vector (s_1, s_2, s_3) corresponds to the initial BFS $(x_1^1, x_2^1, s_1^1, s_2^1, s_3^1)^T = (0; 0; 1{,}500; 1{,}200; 500)^T$, which corresponds to the point $x^1 = (x_1^1, x_2^1)^T = (0, 0)^T$ in the x_1, x_2-space in Figure 3 of the feasible region for this problem.

A nonbasic variable is eligible to enter this basic vector if its updated objective coefficient (i.e., coefficient in the objective row in the canonical tableau) has the appropriate sign to improve the objective value (positive for maximization, negative for minimization). If no nonbasic variables are eligible to enter the present feasible basic vector, the present BFS is an optimum solution to the problem, and the method terminates.

In this tableau, both nonbasic variables x_1, x_2 are eligible to enter the basic vector, among them we selected x_1 as the entering variable, and its column vector in the present canonical tableau becomes the pivot column for this pivot step. If no positive entries are among the constraint rows in the pivot column, the objective function is unbounded (unbounded above if the original problem is a maximization problem, or unbounded below if it is a minimization problem) on the feasible region, and again the method terminates.

If unbounded termination did not occur, the dropping basic variable that the entering variable will replace is determined using the primal simplex minimum ratio test to guarantee that the next basic vector will also remain feasible. For this in each row in which the pivot column has a positive entry, the ratio of the updated RHS constant in that row divided by the entry in the pivot column is computed. The smallest of these ratios is called the minimum ratio, and a row in which it occurs is selected as the pivot row for the pivot operation, and the present basic variable in that row is the dropping variable that will be replaced by the entering variable in the next basic vector.

Table 5. Tableau after the pivot step.

BV	x_1	x_2	s_1	s_2	s_3	$-z$	RHS
s_1	0	−1	1	0	−2	0	500
s_2	0	1	0	1	−1	0	700
x_1	1	0	0	0	1	0	500
$-z$	0	10	0	0	−15	1	−7,500

It is s_3 here, hence the row in which s_3 is basic; Row 3 is the pivot row for this pivot step. Table 5 is the canonical tableau with respect to the basic vector $(s_1, s_2, x_1)^T$ obtained after this pivot step. Its BFS corresponds to the extreme point solution $x^2 = (x_1^2, x_2^2)^T = (500, 0)^T$ in the x_1, x_2-space of Figure 3; it is an adjacent extreme point of x^1. Thus, in this pivot step, the primal simplex method has moved from x^1 to x^2 along the edge of the feasible region joining them, increasing the objective value from 0 to \$7,500 in this process. The method continues from x^2 in the same way.

Each step of the simplex method requires the updating of the basis inverse as the basis changes in one column. Because the method follows a path along the edges (one-dimensional boundary faces or corners) of the set of feasible solutions of the LP, it is classified as a *one-dimensional boundary method*.

4.3. Introduction to Earlier Interior Point Methods for LP

In the early 1980s, Karmarkar pioneered a new method for LP, an interior point method (Karmarkar [16]). Claims were made that this method would be many times faster than the simplex method for solving large-scale sparse LPs; and these claims attracted researchers' attention. His work attracted worldwide attention, not only from operations researchers, but also from scientists in other areas. I will relate a personal experience. When news of his work broke out in world press, I was returning from Asia. The person sitting next to me on the flight was a petroleum geologist. When he learned that I am on the OR faculty at Michigan, he asked me excitedly, "I understand that an OR scientist from India at Bell Labs made a discovery that is going to revolutionize petroleum exploration. Do you know him?!"

In talks at that time on his algorithm, Karmarker repeatedly emphasized the following points: (I) The boundary of a convex polyhedron with its faces of varying dimensions has a highly complex combinatorial structure. Any method that operates on the boundary or close to the boundary will get caught up in this combinatorial complexity, and there is a limit on improvements we can make to its efficiency. (II) Methods that operate in the central portion of the feasible region in the direction of descent of the objective function have the ability to take longer steps toward the optimum before being stopped by the boundary and, hence, have the potential of being more efficient than boundary methods for larger problems. (III) From an interior point, one can move in any direction locally without violating feasibility; hence, powerful methods of unconstrained optimization can be brought to bear on the problem.

Researchers saw the validity of these arguments, and his talks stimulated a lot of work on these methods that stay "away" from the boundary. In the tidal wave of research that ensued, many different classes of interior point methods have been developed for LP, and have extended to wider classes of problems including convex quadratic programming, the monotone linear complementarity problem, and semidefinite programming problems.

4.3.1. Definition of an Interior Feasible Solution and How to Modify the Problem to Have an Initial Interior Feasible Solution Available. In LP literature, an *interior feasible solution* (also called *strictly feasible solution*) to an LP model is defined to be a feasible solution at which all inequality constraints, including bound restrictions on individual variables in the model, are satisfied as strict inequalities but any equality constraints in the model are satisfied as equations. Most interior point methods need an initial interior feasible solution to start the method. If an interior feasible solution to the model is not available, the problem can be modified by introducing one artificial variable using the big-M strategy into a Phase I problem for which an initial interior feasible solution is readily available. We show these modifications first. Suppose the problem to be solved is in the form:

$$\begin{array}{ll} \text{Minimize} & cx \\ \text{subject to} & Ax \geq b \end{array}$$

where A is a matrix of order $m \times n$. For LPs in this form, typically $m \geq n$. Introducing the nonnegative artificial variable x_{n+1}, the Phase I modification of the original problem is

$$\begin{array}{ll} \text{Minimize} & cx + Mx_{n+1} \\ \text{subject to} & Ax + ex_{n+1} \geq b \\ & x_{n+1} \geq 0 \end{array}$$

where $e = (1, \ldots, 1)^T \in R^m$, and M is a positive number significantly larger than any other number in the problem. Let $x^0_{n+1} > \max\{0, b_1, b_2, \ldots, b_m\}$. Then $(0, \ldots, 0, x^0_{n+1})^T$ is an interior feasible solution of the Phase I modification, which is in the same form as the original problem. If the original problem has an optimum solution and M is sufficiently large, then the artificial variable x_{n+1} will be 0 at an optimum solution of the Phase I modification.

Now suppose the original problem is in the form:

$$\begin{array}{ll} \text{Minimize} & cx \\ \text{subject to} & Ax = b \\ & x \geq 0 \end{array}$$

where A is a matrix of order $m \times n$. For LPs in this form, typically $n > m$, and an interior feasible solution is strictly > 0. Select an arbitrary vector $x^0 \in R^n$, $x^0 > 0$; generally, one chooses $x^0 = (1, \ldots, 1)^T$, the n-vector of all ones. If x^0 happens to be feasible to the problem, it is an interior feasible solution, done. Otherwise, let $A_{.n+1} = b - Ax^0$. The Phase I modification including the nonnegative artificial variable x_{n+1} is

$$\begin{array}{ll} \text{Minimize} & cx + Mx_{n+1} \\ \text{subject to} & Ax + A_{.n+1}x_{n+1} = b \\ & x, x_{n+1} \geq 0. \end{array}$$

It is easily confirmed that (x^0, x^0_{n+1}), where $x^0_{n+1} = 1$ is an interior feasible solution of the Phase I problem, which is in the same form as the original problem. Again, if the original problem has an optimum solution and M is sufficiently large, then the artificial variable x_{n+1} will be 0 at an optimum solution of the Phase I modification.

Similar modifications can be made to a general LP in any form, to get a Phase I modification in the same form with an interior feasible solution.

4.3.2. The Structure of the General Step in Interior Point Methods. Assume that the problem being solved is a minimization problem. All interior point methods start with a known interior feasible solution x^0 say, and generate a descent sequence of interior feasible solutions $x^0, x^1, \ldots$. Here, a descent sequence means a sequence along which either the objective value or some other measure of optimality strictly decreases. The general step in all the interior point methods has the following structure:

4.3.3. General Step.

Substep 1. Let x^r be the current interior feasible solution. Generate a search direction d^r at x^r, a descent direction.

Substep 2. Compute the maximum step length θ_r, the maximum value of λ that keeps $x^r + \lambda d^r$ feasible to the original problem. This is like the minimum ratio computation in the simplex method. Determine the *step length fraction parameter* α_r, $0 < \alpha_r < 1$, and take $x^{r+1} = x^r + \alpha_r \theta_r d^r$. With x^{r+1} as the next interior feasible solution, go to the next step.

The various methods differ on whether they work on the primal system only, dual system only, or the system consisting of the primal and dual systems together; on the strategy used to select the search direction d^r; and on the choice of the step length fraction parameter.

To give an idea of the main strategies used by interior point methods to select the search directions, we will discuss the two most popular interior point methods.

The first is the first interior point method discussed in the literature, the primal affine scaling method (Dikin [8]), which predates Karmarkar's work but did not attract much attention until after Karmarkar popularized the study of interior point methods. This method works on the system of constraints in the original problem (primal) only. To get the search direction at the current interior feasible solution x^r, this method creates an ellipsoid $\bar{E}_r$ centered at x^r inside the feasible region of the original LP. Minimizing the objective function over $\bar{E}_r$ is an easy problem, its optimum solution $\bar{x}^r$ can be computed directly by a formula. The search direction in this method at x^r is then the direction obtained by joining x^r to $\bar{x}^r$.

The second method is a central path-following primal-dual interior point method. It works on the system of constraints of both the primal and dual together. In this method, the search directions used are modified Newton directions for solving the optimality conditions. The class of path-following primal-dual methods evolved out of the work of many authors including Bayer and Lagarias [1], Güler et al. [12], Kojima et al. [17], McLinden [19], Meggiddo [20], Mehrotra [21], Mizuno et al. [23], Monteiro and Adler [24], Sonnevend et al. [38], and others. For a complete list of references to these and other authors see the list of references in Saigal [36], Wright [43], and Ye [44].

4.4. The Primal Affine Scaling Method

This method is due to Dikin [8]. We describe the method when the original LP is in the following standard form:

$$\begin{array}{ll} \text{Minimize} & cx \\ \text{subject to} & Ax = b \\ & x \geq 0 \end{array}$$

where A is of order $m \times n$ and rank m. Let x^0 be an available interior feasible solution, i.e., $Ax^0 = b$ and $x^0 > 0$ for initiating the method. The method generates a series of interior feasible solutions $x^0, x^1, \ldots$. We will discuss the general step.

4.4.1. Strategy of the General Step. Let $x^r = (x_1^r, \ldots, x_n^r)^T$ be the current interior feasible solution. The method creates an ellipsoid with x^r as center inside the feasible region of the original LP. It does this by replacing the nonegativity restrictions "$x \geq 0$" by "$x \in E_r = \{x\colon \sum_{i=1}^n ((x_i - x_i^r)/(x_i^r))^2 \leq 1\}$." E_r is an ellipsoid in R^n with its center at x^r. The ellipsoidal approximating problem is then

$$\begin{array}{ll} \text{Minimize} & cx \\ \text{subject to} & Ax = b \\ & \displaystyle\sum_{i=1}^{n} ((x_i - x_i^r)/(x_i^r))^2 \leq 1. \end{array}$$

It can be shown that $E_r \subset \{x\colon x \geq 0\}$. The intersection of E_r with the affine space defined by the system of equality constraints $Ax = b$ is an ellipsoid $\bar{E}_r$ with center x^r inside the feasible region of the original LP. The ellipsoidal approximating problem given above is the problem of minimizing the objective function cx over this ellipsoid $\bar{E}_r$. Its optimum solution $\bar{x}^r = (\bar{x}_j^r)$ can be computed by the formula:

$$\bar{x}^r = x^r - [X_r P_r X_r c^T]/(||P_r X_r c^T||) = x^r - [X_r^2 s^r]/(||X_r s^r||)$$

where $||\cdot||$ indicates the Euclidean norm, and

X_r = diag$(x_1^r, \ldots, x_n^r)$, the diagonal matrix of order n with diagonal entries $x_1^r, \ldots, x_n^r$ and off-diagonal entries 0,
I = unit matrix of order n,
$P_r = (I - X_r A^T (A X_r^2 A^T)^{-1} A X_r)$, a projection matrix,
$y^r = (A X_r^2 A^T)^{-1} A X_r^2 c^T$, known as the *tentative dual solution* corresponding to the current interior feasible solution x^r,
$s^r = c^T - A^T y^r$, *tentative dual slack vector* corresponding to x^r.

It can be shown that if $\bar{x}^r_j = 0$ for at least one j, then $\bar{x}^r$ is an optimum solution of the original LP, and the method terminates. Also, if the tentative dual slack vector s^r is ≤ 0, then the objective value is unbounded below in the original LP, and the method terminates. If these termination conditions are not satisfied, then the search direction at x^r is

$$d^r = \bar{x}^r - x^r = -(X_r^2 s^r)/(||X_r s^r||),$$

known as the *primal affine scaling direction* at the primal interior feasible solution x^r. Because both $x^r, \bar{x}^r$ are feasible to the original problem, we have $Ax^r = A\bar{x}^r = b$, hence, $Ad^r = 0$. So, d^r is a descent feasible direction for the primal along which the primal objective value decreases. The maximum step length θ_r that we can move from x^r in the direction d^r is the maximum value of λ that keeps $x^r_j + \lambda d^r_j \geq 0$ for all j. It can be verified that this is

∞ if $s^r \leq 0$ (this leads to the unboundedness condition stated above); and if $s^r \not\leq 0$, it is equal to

$\theta_r =$ $\min\{(||X_r s^r||)/(x^r_j s^r_j)$: over j such that $s^r_j > 0\}$.

It can be verified that $\theta_r = 1$ if $\bar{x}^r_j = 0$ for some j (in this case, $\bar{x}^r$ is an optimum solution of the original LP as discussed above). Otherwise, $\theta_r > 1$. In this case, the method takes the next iterate to be $x^{r+1} = x^r + \alpha\theta_r d^r$ for some $0 < \alpha < 1$. Typically, $\alpha = 0.95$ in implementations of this method. This α is the *step length fraction parameter*. Then, the method moves to the next step with x^{r+1} as the current interior feasible solution. Here is a summary statement of the general step in this method.

4.4.2. General Step.

Substep 1. Let $x^r = (x^r_1, \ldots, x^r_n)^T$ be the current interior feasible solution of the problem. Let $X_r = \text{diag}(x^r_1, \ldots, x^r_n)$.

Substep 2. Compute the tentative dual solution $y^r = (AX_r^2 A^T)^{-1} AX_r^2 c^T$, the tentative dual slack $s^r = c^t - A^T y^r$, and the primal affine scaling search direction at x^r, which is $d^r = -(X_r^2 s^r)/(||X_r s^r||)$.

If $s^r \leq 0$, $\{x^r + \lambda d^r\colon \lambda \geq 0\}$ is a feasible half-line for the original problem along which the objective function $cx \to -\infty$ as $\lambda \to +\infty$, terminate.

Substep 3. If $s^r \not\leq 0$, compute the maximum step length that we can move from x^r in the direction d^r, this is the maximum value of λ that keeps $x^r_j + \lambda d^r_j \geq 0$ for all j. It is $\theta_r = \min\{(||X_r s^r||)/(x^r_j s^r_j)$: over j such that $s^r_j > 0\}$. If $\theta_r = 1$, $x^r + d^r$ is an optimum solution of the original LP, terminate.

Otherwise let $x^{r+1} = x^r + \alpha d^r$ for some $0 < \alpha < 1$ (typically $\alpha = 0.95$). With x^{r+1} as the current interior feasible solution, go to the next step.

Under some minor conditions, it can be proved that if the original problem has an optimum solution, then the sequence of iterates x^r converges to a strictly complementary optimum solution, and that the objective value cx^r converges at a linear or better rate. Also, if the step length fraction parameter α is $< 2/3$, then the tentative dual sequence y^r converges to the analytic center of the optimum dual solution set. For proofs of these results and a complete discussion of the convergence properties of this method, see Murty [26]. So far, this method has not been shown to be a polynomial time method.

Versions of this method have been developed for LPs in more general forms, such as the bounded variable form and the form in which the LP consists of some unrestricted variables as well. When the original LP has unrestricted variables, instead of an ellipsoid, the method creates a hyper-cylinder with an elliptical cross section inside the feasible region centered at the current interior feasible solution. The point minimizing the objective function over this hyper-cylinder can also be computed directly by a formula, and other features of the method remain essentially similar to the above.

A version of this method that works on the constraints in the dual problem only (instead of those of the primal) has also been developed; this version is called the *dual affine scaling*

method. There is also a *primal-dual affine scaling method* that works on the system consisting of both the primal and dual constraints together; search directions used in this version are based on Newton directions for the system consisting of the complementary slackness conditions.

4.5. Primal-Dual Interior Point Methods for LP

The central path following primal-dual interior point methods are some of the most popular methods for LP. They consider the primal LP:

$$\text{minimize } c^T x, \text{ subject to } Ax = b, x \geq 0$$
$$\text{and its dual in which the constraints are: } A^T y + s = c, s \geq 0,$$

where A is a matrix of order $m \times n$ and rank m. The system of primal and dual constraints put together is

$$\begin{aligned} Ax &= b \\ A^T y + s &= c \\ (x, s) &\geq 0. \end{aligned} \tag{4}$$

A feasible solution (x, y, s) to (4) is called an *interior feasible solution* if $(x, s) > 0$. Let $\mathcal{F}$ denote the set of all feasible solutions of (4), and $\mathcal{F}^0$ the set of all interior feasible solutions. For any $(x, y, s) \in \mathcal{F}^0$, define $X = \text{diag}(x_1, \ldots, x_n)$, the square diagonal matrix of order n with diagonal entries $x_1, \ldots, x_n$; and $S = \text{diag}(s_1, \ldots, s_n)$.

For each $j = 1$ to n, the pair (x_j, s_j) is known as the *jth complementary pair of variables* in these primal-dual pair of problems. The *complementary slackness conditions* for optimality in this pair of problems are: the product $x_j s_j = 0$ for each $j = 1$ to n; i.e., $XSe = 0$ where e is a vector of all ones. Because each product is ≥ 0, these conditions are equivalent to $x^T s = 0$.

4.5.1. The Central Path. The central path, $\mathcal{C}$, for this family of primal-dual path-following methods is a curve in $\mathcal{F}^0$ parametrized by a positive parameter $\tau > 0$. For each $\tau > 0$, the point $(x^\tau, y^\tau, s^\tau) \in \mathcal{C}$ satisfies: $(x^\tau, s^\tau) > 0$ and

$$\begin{aligned} A^T y^\tau + s^\tau &= c^T \\ Ax^\tau &= b \\ x_j^\tau s_j^\tau &= \tau, \quad j = 1, \ldots, n. \end{aligned}$$

If $\tau = 0$, the above equations define the optimality conditions for the LP. For each $\tau > 0$, the solution (x^τ, y^τ, s^τ) is unique, and as τ decreases to 0, the central path converges to the center of the optimum face of the primal-dual pair of LPs.

4.5.2. Optimality Conditions. From optimality conditions, solving the LP is equivalent to finding a solution (x, y, s) satisfying $(x, s) \geq 0$, to the following system of $2n + m$ equations in $2n + m$ unknowns:

$$F(x, y, s) = \begin{bmatrix} A^T y + s - c \\ Ax - b \\ XSe \end{bmatrix} = 0. \tag{5}$$

This is a nonlinear system of equations because of the last equation.

4.5.3. Selecting the Directions to Move. Let the current interior feasible solution be $(\bar{x}, \bar{y}, \bar{s})$. So, $(\bar{x}, \bar{s}) > 0$. Also, the variables in y are unrestricted in sign in the problem.

Primal-dual path-following methods try to follow the central path $\mathcal{C}$ with τ decreasing to 0. For points on $\mathcal{C}$, the value of τ is a measure of closeness to optimality; when it decreases to 0, we are done. Following $\mathcal{C}$ with τ decreasing to 0 keeps all the complementary pair products $x_j s_j$ equal and decreasing to 0 at the same rate.

However, there are two difficulties for following $\mathcal{C}$. One is that it is difficult to get an initial point on $\mathcal{C}$ with all the $x_j s_j$ equal to each other, the second is that $\mathcal{C}$ is a nonlinear curve. At a general solution $(x, y, s) \in \mathcal{F}^0$, the products $x_j s_j$ will not be equal to each other; hence, the parameter $\mu = (\sum_{j=1}^{n} x_j s_j)/n = x^T s/n$, the *average complementary slackness violation measure*, is used as a measure of optimality for them. Because path-following methods cannot exactly follow $\mathcal{C}$, they stay within a loose but well-defined neighborhood of $\mathcal{C}$ while steadily reducing the optimality measure μ to 0.

Staying explicitly within a neighborhood of $\mathcal{C}$ serves the purpose of excluding points (x, y, s) that are too close to the boundary of $\{(x, y, s)\colon x \geq 0, s \geq 0\}$ to make sure that the lengths of steps toward optimality remain long.

To define a neighborhood of the central path, we need a measure of deviation from centrality; this is obtained by comparing a measure of deviation of the various $x_j s_j$ from their average μ to μ itself. This leads to the measure

$$(||(x_1 s_1, \ldots, x_n s_n)^T - \mu e||)/\mu = (||XSe - \mu e||)/\mu$$

where $||\cdot||$ is some norm. Different methods use neighborhoods defined by different norms.

The parameter θ is used as a bound for this measure when using the Euclidean norm. A commonly used neighborhood based on the Euclidean norm $||\cdot||_2$, called the 2-norm neighborhood, defined by

$$\mathcal{N}_2(\theta) = \{(x, y, s) \in \mathcal{F}^0\colon ||XSe - \mu e||_2 \leq \theta\mu\}$$

for some $\theta \in (0, 1)$. Another commonly used neighborhood based on the ∞-norm is the $\mathcal{N}_{-\infty}(\gamma)$, defined by

$$\mathcal{N}_{-\infty}(\gamma) = \{(x, y, s) \in \mathcal{F}^0\colon x_j s_j \geq \gamma\mu,\ j = 1, \ldots, n\}$$

parametrized by the parameter $\gamma \in (0, 1)$. This is a one-sided neighborhood that restricts each product $x_j s_j$ to be at least some small multiple γ of their average μ. Typical values used for these parameters are $\theta = 0.5$, and $\gamma = 0.001$. By keeping all iterates inside one or the other of these neighborhoods, path-following methods reduce all $x_j s_j$ to 0 at about the same rates.

Since the width of these neighborhoods for a given μ depends on μ, these neighborhoods are conical (like a *horn*), are wider for larger values of μ, and become narrow as $\mu \to 0$.

Once the direction to move from the current point $(\bar{x}, \bar{y}, \bar{s})$ is computed, we may move from it only a small step length in that direction, and because $(\bar{x}, \bar{s}) > 0$, such a move in any direction will take us to a point that will continue satisfying $(x, s) > 0$. So, in computing the direction to move at the current point, the nonnegativity constraints $(x, s) \geq 0$ can be ignored. The only remaining conditions to be satisfied for attaining optimality are the equality conditions (5). So, the direction-finding routine concentrates only on trying to satisfy (5) more closely.

Ignoring the inactive inequality constraints in determining the direction to move at the current point is the main feature of *barrier methods* in nonlinear programming, hence, these methods are also known as barrier methods.

Equation (5) is a square system of nonlinear equations ($2n + m$ equations in $2n + m$ unknowns, it is nonlinear because the third condition in (5) is nonlinear). Experience in nonlinear programming indicates that the best directions to move in algorithms for solving nonlinear equations are either the Newton direction or some modified Newton direction. So, this method uses a modified Newton direction to move. To define that, a centering parameter $\sigma \in [0, 1]$ is used. Then, the direction for the move denoted by $(\Delta x, \Delta y, \Delta s)$ is the solution to the following system of linear equations

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -XSe + \sigma\mu e \end{pmatrix} \tag{6}$$

where 0 in each place indicates the appropriate matrix or vector of zeros, I the unit matrix of order n, and e indicates the column vector of order n consisting of all ones.

If $\sigma = 1$, the direction obtained will be a centering direction, which is a Newton direction toward the point (x^μ, y^μ, s^μ) on $\mathcal{C}$ at which the products $x_j s_j$ of all complementary pairs in this primal-dual pair of problems are $= \mu$. Moving in the centering direction helps to move the point toward $\mathcal{C}$, but may make little progress in reducing the optimality measure μ. But in the next iteration, this may help to take a relatively long step to reduce μ. At the other end, the value $\sigma = 0$ gives the standard Newton direction for solving (5). Many algorithms choose σ from the open interval $(0, 1)$ to trade off between twin goals of reducing μ and improving centrality.

We now describe two popular path-following methods.

4.5.4. The Long-Step Path-Following Algorithm (LPF). LPF generates a sequence of iterates in the neighborhood $\mathcal{N}_{-\infty}(\gamma)$, which for small values of γ (for example, $\gamma = 0.001$) includes most of the set of interior feasible solutions $\mathcal{F}^0$. The method is initiated with an $(x^0, y^0, s^0) \in \mathcal{F}^0$. In each step, the method chooses the centering parameter σ between two selected limits $\sigma_{\min}, \sigma_{\max}$ where $0 < \sigma_{\min} < \sigma_{\max} < 1$. The neighborhood-defining parameter γ is selected from $(0, 1)$. Here is the general step in this algorithm.

4.5.5. General Step k. Let (x^k, y^k, s^k) be the current interior feasible solution, and $\mu_k = (x^k)^T s^k / n$ the current value of the optimality measure corresponding to it. Choose $\sigma_k \in [\sigma_{\min}, \sigma_{\max}]$. Find the direction $(\Delta x^k, \Delta y^k, \Delta s^k)$ by solving

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{pmatrix} \begin{pmatrix} \Delta x^k \\ \Delta y^k \\ \Delta s^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -X^k S^k e + \sigma_k \mu_k e \end{pmatrix}. \tag{7}$$

Find $\alpha_k =$ the largest value of $\alpha \in [0, 1]$ such that $(x^k, y^k, s^k) + \alpha(\Delta x^k,\ \Delta y^k,\ \Delta s^k) \in \mathcal{N}_{-\infty}(\gamma)$.

Setting $(x^{k+1}, y^{k+1}, s^{k+1}) = (x^k, y^k, s^k) + \alpha_k(\Delta x^k, \Delta y^k, \Delta s^k)$ as the new current interior feasible solution, go to the next step.

4.5.6. The Predictor-Corrector Path-Following Method (PC). Path-following methods have two goals: one to improve centrality (closeness to the central path while keeping optimality measure unchanged) and the other to decrease the optimality measure μ. The PC method takes two different steps alternately to achieve each of these twin goals. The PC uses two $\mathcal{N}_2$ neighborhoods nested one inside the other. They are $\mathcal{N}_2(\theta_1), \mathcal{N}_2(\theta_2)$ for selected $0 < \theta_1 < \theta_2 < 1$. For example $\theta_1 = 0.25, \theta_2 = 0.5$. In some versions of this method, values of θ larger than 1 are also used successfully.

Every second step in this method is a "predictor" step; its starting point will be in the inner neighborhood. The direction to move in this step is computed by solving the system (7) corresponding to the current solution with the value of $\sigma = 0$. The step length in this step is the largest value of α that keeps the next point within the outer neighborhood. The gap between the inner and outer neighborhoods is wide enough to allow this step to make significant progress in reducing μ.

The step taken after each predictor step is a "corrector" step, its starting point will be in the outer neighborhood. The direction to move in this step is computed by solving the system (7) corresponding to the current solution with the value of $\sigma = 1$. The step length in this step is $\alpha = 1$, which takes it back inside the inner neighborhood to prepare for the next predictor step.

It has been shown that the sequence of interior feasible solutions obtained in this method converges to a point in the optimum face. All these path-following methods have been shown to be polynomial time algorithms.

Each step of these interior point methods requires a full matrix inversion, a fairly complex task in solving large-scale problems, this involves much more work than a step of the simplex method. But the number of steps required by these interior point methods is smaller than the number of steps needed by the simplex method.

5. Gravitational Methods with Small Balls (Higher-Dimensional Boundary Methods)

Chang [2], pointed out that the path taken by the simplex algorithm to solve an LP can be interpreted as the path of a point ball falling under the influence of a gravitational force inside a thin tubular network of the one-dimensional skeleton of the feasible region in which each vertex is open to all the edges incident at it. See Figure 4 for a two-dimensional illustration.

Murty [27, 28] introduced newer methods for LP based on the principle of the gravitational force, Chang and Murty [3] extended this further. They consider an LP in the form

$$\begin{array}{ll} \text{maximize} & \pi b \\ \text{subject to} & \pi A = c, \quad \pi \geq 0 \end{array} \tag{8}$$

where A is a matrix of order $m \times n$, $\pi \in R^m$ is the row vector of primal variables. As explained in §1, for problems in this form, typically $n \leq m$. Its dual is

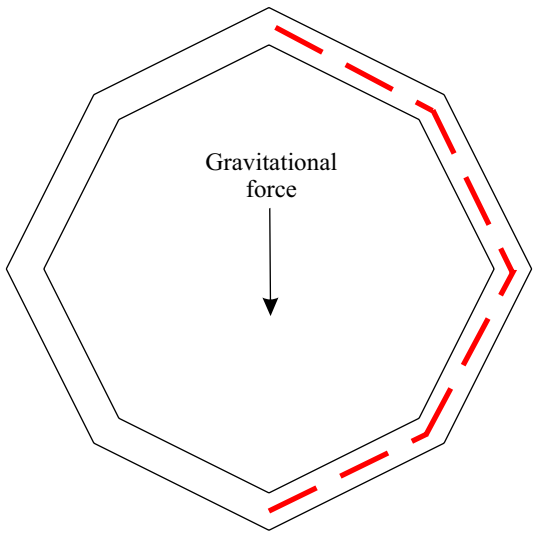
$$\begin{array}{ll} \text{minimize} & z(x) = cx \\ \text{subject to} & Ax \geq b \end{array} \tag{9}$$

where $x \in R^n$ is the column vector of dual variables.

We use the symbols $A_{i.}, A_{.j}$ to denote the ith row vector, jth column vector of the matrix A. We assume that the rows of A have all been normalized so that $||A_{i.}|| = 1$ for all i, where $||\cdot||$ is the Euclidean norm. We also assume that $c \neq 0$ and that it is normalized so that $||c|| = 1$.

The method is applied on (9). We denote its feasible region $\{x\colon Ax \geqq b\}$ by K, and its interior $\{x\colon Ax > b\}$ by K^0. The method needs an initial *interior point* $x^0 \in K^0$. It introduces

FIGURE 4. The gravitational interpretation of the simplex method.



Notes. The dashed lines indicate the path taken by a point ball beginning at the top vertex inside a tubular network for the edges of the feasible region of an LP under the gravitational force pulling it toward the optimum.

a spherical drop (we will refer to it as the *drop* or the *ball*) of small radius with center x^0 lying completely in the interior of K, and traces the path of its center as the drop falls under a gravitational force pulling everything in the direction $-c^T$. The drop cannot cross the boundary of K, so after an initial move in the direction $-c^T$, it will be blocked by the face of K that it touches; after which it will start rolling down along the faces of K of varying dimensions. Hence, the center of the drop will follow a piecewise linear descent path completely contained in the interior of K, but because the drop's radius is small, the center remains very close to the boundary of K after the first change in direction in its path. Therefore, the method is essentially a boundary method. However, unlike the simplex method that follows a path strictly along the one-dimensional boundary of K, this method is a *higher-dimensional boundary method* in which the path followed remains very close to faces of K of varying dimensions. See Figures 5 and 6 for two-, three-dimensional illustrations. After a finite number of changes in the direction of movement, the drop will reach the lowest possible point in the direction $-c^T$ that it can reach within K and then halt. If the radius of the drop is sufficiently small, the touching constraints (i.e., those whose corresponding facets of K are touching the ball) in (9) at this final halting position will determine an actual optimum solution of the LP (8). If its radius is not small enough, the direction-finding step in the method at the final halting position with center x^* yields a feasible solution $\tilde{\pi}$ of (8), and the optimum objective value in (8) lies in the interval $[\tilde{\pi}b, cx^*]$. Then the radius of the drop is reduced and the method continues the same way. In Chang and Murty [3], finite termination of the method to find an optimum solution has been proved.

The algorithm consists of one or more stages. In each stage, the diameter of the ball remains unchanged and consists of a series of iterations. Each iteration consists of two steps: a step that computes the gravitational direction for moving the entire ball, and a step in which the step length for the move is computed and the ball moved. The stage ends when the ball cannot move any further and halts. In the very first iteration of each stage, the ball will be strictly in the interior of K without touching any of the facets of K. In subsequent iterations, it will always be touching one or more facets of K. We will now describe a general stage.

5.1. A Stage in the Gravitational Method

5.1.1. First Iteration. Let x^0 be the present interior feasible solution. The largest sphere we can construct within K with x^0 as center has radius $= \min\{A_{i.}x^0 - b_i\colon\ i = 1 \text{ to } m\}$. Let $B(x^0, \epsilon) = \{x\colon\ ||x - x^0|| \leq \epsilon\}$ be the present ball. In this iteration, we will have $0 < \epsilon < \min\{A_{i.}x^0 - b_i\colon\ i = 1 \text{ to } m\}$, so $B(x^0, \epsilon)$ is not touching any of the facets of K.

FIGURE 5. A two-dimensional polytope and its faces on which the ball rolls down (dashed path) to the optimum.

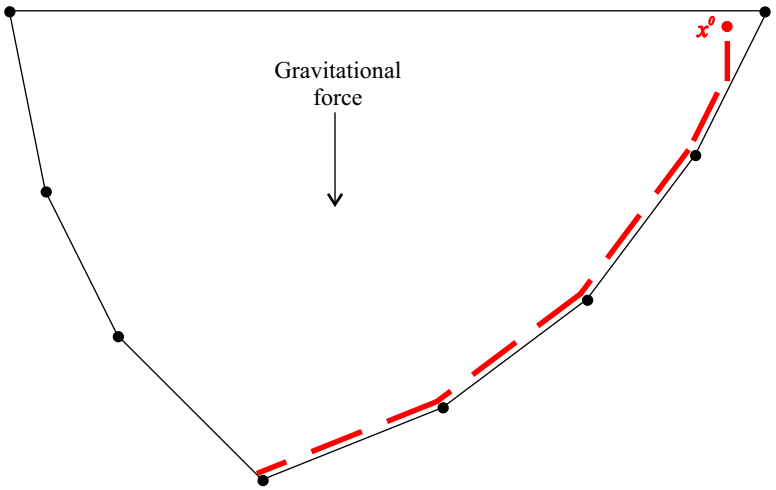
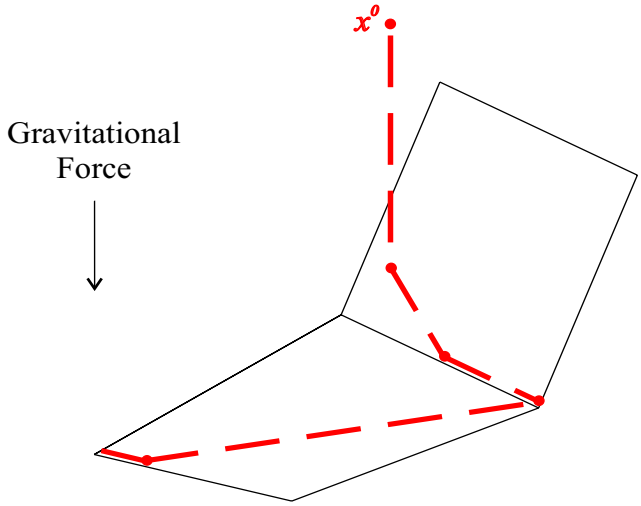


FIGURE 6. The ball rolling (dashed path, with dots indicating where its direction changes) inside a three-dimensional polyhedron.



Note. Only the faces along which it rolls to the optimum are shown.

In this iteration, the entire ball is moved in the direction $-c^T$. The step length is the maximum value of λ satisfying $A_{i.}(x^0 - \lambda c^T) - b_i \geq \epsilon$ for all i. So, it is

$$\gamma = \begin{cases} \infty & \text{if } A_{i.}c^T \leq 0 \quad \text{for all } i \\ \min\left\{ \dfrac{A_{i.}x^0 - b_i - \epsilon}{A_{i.}c^T} \colon\ i \text{ such that } A_{i.}c^T > 0 \right\} & \text{otherwise.} \end{cases}$$

If $\gamma = \infty$, the objective function in (9) is unbounded below on its feasible set, and (8) is infeasible, terminate. Otherwise, move the center of the ball from x^0 to $x^1 = x^0 - \gamma c^T$. With the new position $B(x^1, \epsilon)$ of the ball, go to the next iteration.

5.2. General Iteration $r \geq 1$

Let x^{r-1} be the current interior feasible solution and $B(x^{r-1}, \epsilon)$ the present ball. Let

$$\begin{aligned} J(x^{r-1}, \epsilon) &= \{i \colon A_{i.}x^{r-1} = b_i + \epsilon\}, \text{ the index set of touching constraints for } B(x^{r-1}, \epsilon) \\ Q &= \text{the matrix consisting of rows } A_{i.} \text{ for } i \in J(x^{r-1}, \epsilon) \\ G(x^{r-1}, \epsilon) &= \{y \colon cy < 0, A_{i.}y \geqq 0 \text{ for all } i \in J(x^{r-1}, \epsilon)\}, \text{ the set of descent feasible} \\ &\qquad \text{directions for the ball } B(x^{r-1}, \epsilon). \end{aligned}$$

Step 1. Selecting the gravitational direction at x^{r-1} for moving the entire current ball $B(x^{r-1}, \epsilon)$.

The steepest descent gravitational method (SDGM) developed in Chang and Murty [3] takes this direction to be the steepest direction among all those in $G(x^{r-1}, \epsilon)$. This direction, called the SDGD (steepest descent gravitational direction) at x^{r-1} is the optimum solution of

$$\begin{aligned} \text{Minimize} \quad & cy \\ \text{subject to} \quad & Qy \geqq 0 \\ & 1 - y^T y \geqq 0. \end{aligned} \tag{10}$$

This problem is equivalent to

$$\begin{aligned} \text{Minimize} \quad & (c - \eta Q)(c - \eta Q)^T \\ \text{subject to} \quad & \eta \geqq 0, \end{aligned} \tag{11}$$

which is the same as that of finding the nearest point by Euclidean distance to c in the cone $\text{Rpos}(Q)$ = the nonnegative hull of row vectors of Q. This is a quadratic program, but is expected to be small because its number of variables is equal to the number of touching constraints at x^{r-1}, which is likely to be small. Also, this is a special quadratic program of finding the nearest point to c in a cone expressed as the nonnegative hull of row vectors of a matrix, for which efficient geometric methods are available Murty and Fathi [34], Wilhelmsen [40], and Wolfe [41, 42].

If $\bar{\eta}$ is an optimum solution of (11), let

$$\bar{y}^{r-1} = \begin{cases} 0 & \text{if } \bar{\xi} = (c - \bar{\eta}Q) = 0 \\ -\bar{\xi}^T/||\bar{\xi}|| & \text{otherwise} \end{cases}$$

then $\bar{y}^{r-1}$ is an optimum solution of (10).

If $\bar{\xi} = \bar{y}^{r-1} = 0$, then $G(x^{r-1}, \epsilon) = \varnothing$, implying that the drop $B(x^{r-1}, \epsilon)$ cannot move any further in gravitational descent with gravity pulling everything in the direction of $-c^T$; hence, it halts in the present position, and the method moves to the final step in this stage.

If $\bar{y}^{r-1} \neq 0$, it is selected as the gravitational direction for the ball $B(x^{r-1}, \epsilon)$ to move, and the method goes to Step 2 in this iteration.

Reference [3] also discusses simpler methods for choosing the gravitational direction for the ball $B(x^{r-1}, \epsilon)$ to move, by solving the nearest point problem (11) approximately rather than exactly based on efficient geometric procedures discussed in Karmarkar [16].

Step 2. Step length determination and moving the ball. The maximum step length that the ball $B(x^{r-1}, \epsilon)$ can move in the direction $\bar{y}^{r-1}$ is the maximum value of λ that keeps $A_{i.}(x^{r-1} + \lambda \bar{y}^{r-1}) \geq b_i + \epsilon$ for all $i = 1$ to m. It is

$$\gamma_{r-1} = \begin{cases} \infty & \text{if } A_{i.}\bar{y}^{r-1} \geq 0 \quad \text{for all } i \\ \min\left\{ \dfrac{A_{i.}x^{r-1} - b_i - \epsilon}{-A_{i.}y^{r-1}} \colon i \text{ such that } A_{i.}y^{r-1} < 0 \right\} & \text{otherwise.} \end{cases}$$

If $\gamma_{r-1} = \infty$, the algorithm terminates with the conclusion that the objective function is unbounded below in (9) (in fact, the half-line $\{x^{r-1} + \lambda y^{r-1} \colon \lambda \geq 0\}$ is a feasible half-line in K along which $z \to -\infty$), and (8) is infeasible. If γ^{r-1} is finite, the center of the drop is moved from x^{r-1} to $x^r = x^{r-1} + \gamma_{r-1}y^{r-1}$. With the ball in the new position $B(x^r, \epsilon)$, the method now moves to the next iteration.

The Final Step in a Stage. Suppose the ball halts in some iteration r with the ball in position $B(x^{r-1}, \epsilon)$. $J(x^{r-1}, \epsilon)$ is the index set of touching constraints in this iteration, and let $\bar{\eta}^{r-1}$ be the optimum solution of (11). Then, it can be verified that if we define

$$\bar{\pi}_i = \begin{cases} \bar{\eta}_i^{r-1} & \text{for } i \in J(x^{r-1}, \epsilon) \\ 0 & \text{otherwise,} \end{cases}$$

then $\bar{\pi} = (\bar{\pi}_i)$ is a feasible solution to (8). In this case, both (8) and (9) have optimum solutions, and the optimum objective value z^* in them satisfies $\bar{\pi}b \leq z^* \leq cx^{r-1}$. If the difference $cx^{r-1} - \bar{\pi}b$ is sufficiently small, there are several results in LP theory to obtain an optimum solution to (8) from $\bar{\pi}$ that require a small number of pivot steps. Also, let $F = \{i \colon \bar{\pi}_i > 0\}$, and $E \subset F$ such that $\{A_{i.} \colon i \in E\}$ is a maximal linearly independent subset of $\{A_{i.} \colon i \in F\}$, and $d = (b_i \colon i \in E)$. Let $\hat{x} = x^{r-1} + E^T(EE^T)^{-1}(d - Ex^{r-1})$, the orthogonal projection of x^{r-1} on the flat $\{x \colon A_{i.}x = b_i, i \in E\}$. If $\hat{x}$ is feasible to (9), then it is optimal to (9), and $\bar{\pi}$ is optimal to (8), terminate the algorithm.

Suppose $\hat{x}$ is not feasible to (9), then reduce the radius of the ball to half its present value, and with $B(x^{r-1}, \epsilon/2)$ go to the next stage.

In Chang and Murty [3], finite convergence of this algorithm has been proved. In a computational experiment on LPs with up to 200 variables, an experimental code for this method

performed up to six times faster than versions of simplex method professional software available at that time.

In the simplex method and all the interior point methods discussed earlier, all the constraints in the problem including any redundant constraints play a role in the computations (i.e., pivot steps or matrix inversions) in every step. One of the biggest advantages of the gravitational methods is that, in each step, only a small locally defined set of constraints (these are the touching constraints in that step) play a role in the major computation, and, in particular, redundant constraints can never enter the touching set; therefore, the computational effort in each iteration is significantly less than in other methods.

The radius of the ball is kept small, and after the first move in the direction $-c^T$, the ball keeps rolling on the boundary faces of K of various dimensions, hence, as explained earlier, this method can be classified as a higher-dimensional boundary method. The worst-case complexity of this method when the ball has positive radius that changes over the algorithm has not been established, but Morin et al. [25] showed that the version of the method with a point ball having 0 radius or any fixed radius has exponential complexity in the worst case.

6. A New Predictor-Corrector-Type Interior Point Method Based on a New Simpler Centering Strategy that Can Be Implemented Without Matrix Inversions

We will now discuss a new interior point method developed recently in Murty [30, 33]. We have seen that in the gravitational methods discussed in §5 using balls of small radius, the path traced by the center of the ball—even though it is strictly in the interior of the set of feasible solutions of the LP—essentially rolls very close to the boundary, hence, making the method behave like a boundary method rather than a truely interior point method.

To make the gravitational method follow a path truely in the central part of the feasible region and benefit from the long steps toward optimality possible under it, this new method modifies it by using balls of the highest possible radius obtained through a special centering strategy.

In the gravitational methods of §5, the majority of the work goes into computing the descent directions for the ball to move. In the new method, however, much of the work is in centering steps. The method considers LPs in the form

$$\begin{array}{rl} \text{Minimize} & z(x) = cx \\ \text{subject to} & Ax \geq b \end{array} \tag{12}$$

where A is a matrix of order $m \times n$. In this form, typically $m \geq n$. We let K denote the set of feasible solutions of this LP and $K^0 = \{x\colon Ax > b\}$ its interior. The method needs an initial interior feasible solution $x^0 \in K^0$ to start; if such a solution is not available, the problem can be modified using an artificial variable and the big-M augmentation technique into another one for which an initial interior feasible solution is readily available as explained in §4.3. We assume $c \neq 0$, because otherwise x^0 is already an optimum solution of this LP and 0 is the optimum solution of its dual. We normalize so that $||c|| = ||A_{i.}|| = 1$ for all i, here $A_{i.}$ is the ith row vector of A.

The method consists of a series of iterations, each consisting of two steps: a centering step and a descent step. The first iteration begins with the initial interior feasible solution x^0; subsequent iterations begin with the interior feasible solution obtained at the end of the previous iteration. For any interior feasible solution x, the radius of the largest ball with center at x that can be constructed within K is denoted by

$$\delta(x) = \text{minimum } \{A_{i.}x - b_i\colon i = 1 \text{ to } m\}.$$

Also, in this method, ϵ denotes a small positive tolerance number for "interiorness" (i.e., for $\delta(x)$) for the feasible solution x to be considered an interior feasible solution. We will now describe the steps in a general iteration.

6.1. General Iteration $r+1$

Step 1. Centering. Let x^r be the current interior feasible solution for initiating this iteration. With x^r as center, the largest ball we can construct within K has radius $\delta(x^r)$, which may be too small. To construct a larger ball inside K, this step tries to move the center of the ball from x^r to a better interior feasible solution while keeping the objective value unchanged. So, starting with x^r, it tries to find a new position x for the center of the ball in $K^0 \cap H$ where $H = \{x\colon cx = cx^r\}$ is the objective plane through x^r, to maximize $\delta(x)$. The model for this choice is

$$\begin{array}{rl} \text{Maximize} & \delta \\ \text{subject to} & \delta \le A_{i.}x - b_i, \quad i = 1 \text{ to } m \\ & cx = cx^r. \end{array} \tag{13}$$

This is another LP with variables (δ, x). It may have alternate optimum solutions with different x-vectors, but the optimum value of δ will be unique. If $(\bar{x}^r, \bar{\delta}^r)$ is an optimum solution for it, $\bar{x}^r$ is taken as the new center for the drop, and $\bar{\delta}^r = \delta(\bar{x}^r)$ is the maximum radius for the drop within K^0 subject to the constraint that its center lie on $K^0 \cap H$.

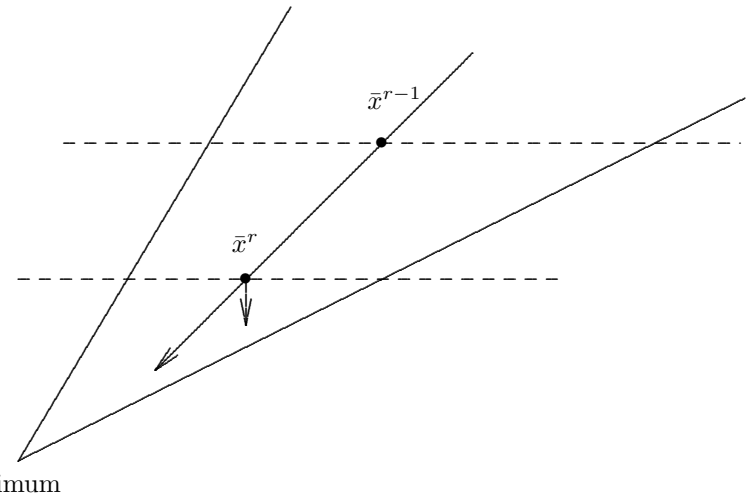
However this itself is another LP; this type of model may have to be solved several times before we get a solution for our original LP, so solving this model (13) exactly will be counterproductive. But (13) has a very special structure; using it, we discuss procedures to get an approximate solution for it later on.

Step 2. Descent move following centering. Let $\bar{x}^r$ denote the center of the ball selected in Step 1. The ball is $B(\bar{x}^r, \delta(\bar{x}^r))$. Unlike the gravitational methods discussed in §5 in which the entire ball is moved, this method does not move the ball $B(\bar{x}^r, \delta(\bar{x}^r))$ at all, but only uses the center $\bar{x}^r$ and its property of being close to the center of $K^0 \cap H$. It takes a step of maximum possible length from $\bar{x}^r$ in a descent direction for cx.

If $r = 0$ (i.e., this is the first iteration in the method), the only descent direction that we have readily available at this time is $-c^T$, and we use that as the direction to move from $\bar{x}^0$.

If $r \ge 1$, besides $-c^T$, we have another descent direction for cx, namely the direction of the *path of centers* (the path of the center of the drop in its descent to the optimum face of (12) in this algorithm) at the current center $\bar{x}^r$, which can be approximated by $\bar{x}^r - \bar{x}^{r-1}$ where $\bar{x}^{r-1}$ was the center of the drop in the previous iteration. See Figure 7.

FIGURE 7. The two descent directions to move in Step 2 when the center is at $\bar{x}^r$ in an iteration.



Notes. One is $\bar{x}^r - \bar{x}^{r-1}$ where $\bar{x}^{r-1}$ is the center in the previous iteration, another is $-c^T$ (here shown as pointing downsouth). The dashed lines are the objective planes in the two iterations.

If $d \in \{-c^T, \bar{x}^r - \bar{x}^{r-1}\}$ is the direction selected for moving from $\bar{x}^r$, we will move in this direction the maximum distance possible while still remaining inside K^0, which is

$$\gamma = \min\left\{ \frac{-A_{i.}\bar{x}^r + b_i + \epsilon}{A_{i.}d} : i \text{ such that } A_{i.}d < 0 \right\}.$$

If $\gamma = \infty$, the objective function is unbounded below in (12), and its dual is infeasible, terminate the algorithm.

If γ is finite, the decrease in the objective value in this move is $|\gamma cd|$. Select the direction d from $\{-c^T, \bar{x}^r - \bar{x}^{r-1}\}$ to be the one that yields the maximum decrease in the objective value in this move. With the point obtained after the move, $x^{r+1} = \bar{x}^r + \gamma d$, go to the next iteration.

6.2. Other Descent Directions

Suppose r iterations have been carried out so far. Then, $\bar{x}^q - \bar{x}^p$ is a descent direction for the objective function in (12) for all $1 \le p < q \le r$. Among all these descent directions, the ones obtained using recent pairs of centers may have useful information about the shape of the feasible region between the objective value at present and at its optimum. So, using a weighted average of these descent directions as the direction to move next (instead of using either $-c^T$ or $\bar{x}^r - \bar{x}^{r-1}$ as discussed above) may help in maximizing the improvement in the objective value in this move. The best weighted average to use for maximum practical effectiveness can be determined using computational experiments.

6.3. Convergence Results

We will summarize the main convergence results on this algorithm under the assumption that centering is carried to optimality in each iteration. Proofs are not given; for them, see Murty [33].

Here, t is a parameter denoting the objective value cx. $t_{\min}, t_{\max}$ denote the minimum and maximum values of cx over K. For any t between $t_{\min}$ and $t_{\max}$, $\delta[t]$ denotes the maximum value of $\delta(x)$ over $x \in K^0 \cap \{x: cx = t\}$; it is the radius of the largest sphere that can be constructed within K with its center restricted to $K^0 \cap \{x: cx = t\}$; it is the optimum value of δ in the LP

$$\begin{aligned} \delta[t] = \text{Maximum value of } & \delta \\ \text{subject to} \quad & \delta - A_{i.}x \le -b_i, \quad i = 1, \ldots, n \\ & cx = t. \end{aligned} \tag{14}$$

The *set of touching constraints at* t is the set of all inequality constraints in (14) satisfied as equations by any of the optimum solutions of (14).

The *essential touching constraint index set at* t is the set $J(t) = \{i: A_{i.}x = b_i + \delta[t]\}$ for every optimum solution $(\delta[t], x)$ of (14)}. The ith constraint in (12), (14) is said to be in the set of essential touching constraints at t if $i \in J(t)$.

We assume that the center selected in the centering strategy is an $x(t)$ satisfying the property that the facets of K touching the ball $B(x(t), \delta[t])$ (the ball with $x(t)$ as center and $\delta[t] = \delta(x(t))$ as radius) are those corresponding to the essential touching constraint set $J(t)$.

6.4. The Path of Centers $\mathcal{P}$

In primal-dual path following interior point algorithms discussed in §4.5, we defined the central path $\mathcal{C}$ in the space of primal-dual variables, parameterized by the parameter τ (the common complementary slackness violation parameter, for points on the central path; this violation is equal in all complementary pairs in this primal-dual pair of LPs). Analogous to

that, we have the path $\{x(t)\colon t_{\max} \geq t \geq t_{\min}\}$ in the space of the variables in the original LP (12) being solved in this algorithm, parameterized by the parameter t denoting the objective function value. We will call this the *path of centers* in this method and denote it by $\mathcal{P}$. We also have the associated path $\{\delta[t]\colon t_{\max} \geq t \geq t_{\min}\}$ of the radii of the balls, which is piecewise linear concave (see Theorem 2 next). Notice the differences. The point on the central path $\mathcal{C}$ is unique for each positive value of the parameter τ. The point $x(t)$ on the path of centers $\mathcal{P}$, however, may not be unique.

Theorem 2. *$\delta[t]$ is a piecewise linear concave function defined over $t_{\min} \leq t \leq t_{\max}$.*

Let $t^* =$ the value of t where $\delta[t]$ attains its maximum value. So, $\delta[t]$ is monotonic increasing as t increases from $t_{\min}$ to t^*, and from t^* it is monotonic decreasing as t increases on to $t_{\max}$.

Theorem 3. *If $J(t)$ remains the same for all $t_1 \leq t \leq t_2$, then $\delta[t]$ is linear in this interval.*

Theorem 4. *For t in the interval $t_{\min}$ to t^*, $x(t)$, an optimum solution of (14), is also an optimum solution of*

$$\begin{array}{ll} \textit{minimize} & cx \\ \textit{subject to} & Ax \geq b + e\delta[t] \end{array}$$

where e is the column vector of all ones of appropriate dimension. And for t in the interval t^* to $t_{\max}$, $x(t)$ is also an optimum solution of

$$\begin{array}{ll} \textit{maximize} & cx \\ \textit{subject to} & Ax \geq b + e\delta[t]. \end{array}$$

Theorem 5. *Suppose for $t_1 \geq t \geq t_2$, the index set of essential touching constraints $J(t)$ does not change. Then, the method will descend from objective value t_1 to t_2 in no more than three iterations.*

Theorem 6. *As t, the value of cx, decreases to $t_{\min}$, the set of essential touching constraints can change at most $2m$ times.*

Theorems 5 and 6 together show that this algorithm is a strongly polynomial algorithm in terms of the number of centering steps, if centering is carried out exactly. So, if the centering steps are carried to good accuracy, these results indicate that this method will have superior computational performance.

6.5. Procedures for Getting Approximate Solutions to Centering Steps Efficiently

Consider the centering step in iteration $r+1$ of the method when x^r is the interior feasible solution at the start of this iteration. We discuss three procedures for solving this step approximately. Procedures 1 and 2 use a series of line searches on $K^0 \cap \{x\colon cx = cx^r\}$. Each line search involves only solving a two-variable linear programming problem, so it can be solved very efficiently without complicated matrix inversions. So, these searches generate a sequence of points that we denote by $\hat{x}^1, \hat{x}^2, \ldots$ in $K^0 \cap \{x\colon cx = cx^r\}$ beginning with $\hat{x}^1 = x^r$, along which $\delta(\hat{x}^s)$ is strictly increasing.

Let $\hat{x}^s$ be the current point in this sequence. Let $T(\hat{x}^s) = \{q\colon q \text{ ties for the minimum in } \{A_{i.}\hat{x}^s - b_i\colon i = 1 \text{ to } m\}\}$. In optimization literature, when considering a line search at $\hat{x}^s$ in the direction P, only moves of positive step length α leading to the point $\hat{x}^s + \alpha P$ are considered. Here, our step length α can be either positive or negative, so even though we mention P as the direction of movement, the actual direction for the move may be either P or $-P$. With $\hat{x}^s + \alpha P$ as the center, the maximum radius of a ball inside K has radius

$$f(\alpha) = \min\{A_{i.}(\hat{x}^s + \alpha P) - b_i\colon i = 1, \ldots, m\}.$$

Because we want the largest ball inside K with its center in $K^0 \cap \{x\colon cx = cx^r\}$, we will only consider directions P satisfying $cP = 0$, and call such a direction P to be a

profitable direction to move at $\hat{x}^s$ if $f(\alpha)$ increases as α changes from 0 to positive or negative values (i.e., $\max\{f(\alpha)$ over $\alpha\}$ is attained at some $\alpha \neq 0$).

unprofitable direction to move at $\hat{x}^s$ if $\max\{f(\alpha)$ over $\alpha\}$ is attained at $\alpha = 0$.

We have the following results.

Result 1. $\hat{x}^s$ is an optimum solution for the centering problem (14) if 0 is the unique feasible solution for the following system in P

$$\begin{aligned} A_{i.}P &\geq 0 \quad \text{for all } i \in T(\hat{x}^s) \\ cP &= 0. \end{aligned} \tag{15}$$

Any nonzero solution to this system is a profitable direction to move at $\hat{x}^s$ for this centering step. Hence, a direction P is a profitable direction to move at $\hat{x}^s$ if $cP = 0$, and all $A_{i.}P$ for $i \in T(\hat{x}^s)$ have the same sign.

Result 2. Suppose P is a profitable direction to move at $\hat{x}^s$, and let $\bar{\alpha}$ denote the value of α that maximizes $f(\alpha)$, and $\bar{\theta} = f(\bar{\alpha})$. Then, $(\bar{\theta}, \bar{\alpha})$ is an optimum solution of the following two-variable LP in which the variables are θ, α.

$$\begin{aligned} \text{Maximize} \quad & \theta \\ \text{subject to} \quad & \theta - \alpha A_{i.}P \leqq A_{i.}\hat{x}^s - b_i \quad 1 = 1, \ldots, m \\ & \theta \geqq 0, \quad \alpha \text{ unrestricted in sign.} \end{aligned} \tag{16}$$

The optimum solution of (16) can be found by applying the simplex algorithm. Transform (16) into standard form. Let $u_1, \ldots, u_m$ denote the slack variables corresponding to the constraints in (16) in this order. Then $(u_1, \ldots, u_{q-1}, \theta, u_{q+1}, \ldots, u_m)$ is a feasible basic vector for this standard form for $q \in T(\hat{x}^s)$. The BFS corresponding to this basic vector for the standard form corresponds to the extreme point $(\delta(\hat{x}^s), 0)$ of (16) in the (θ, α)-space. Starting from this feasible basic vector, the optimum solution of (16) can be found efficiently by the primal simplex algorithm with at most $O(m)$ effort. It may be possible to develop even more efficient ways for finding the optimum value of α in (16); that value is the optimum step length for the move at $\hat{x}^s$ in the profitable direction P.

Using these results, we discuss two procedures for approximating the centering problem (16).

Procedure 1. Getting an Approximate Solution to the Centering Step. Since our goal is to increase the minimum distance of x to each of the facetal hyperplanes of K, this procedure considers only moves in directions perpendicular to the facetal hyperplanes of K; these are the directions $A_{i.}^T$ for $i = 1$ to m. Let $P_{.i} = (I - c^T c)A_{i.}^T$ (where I is the unit matrix of order n); it is the orthogonal projection of $A_{i.}^T$ on $\{x\colon cx = 0\}$.

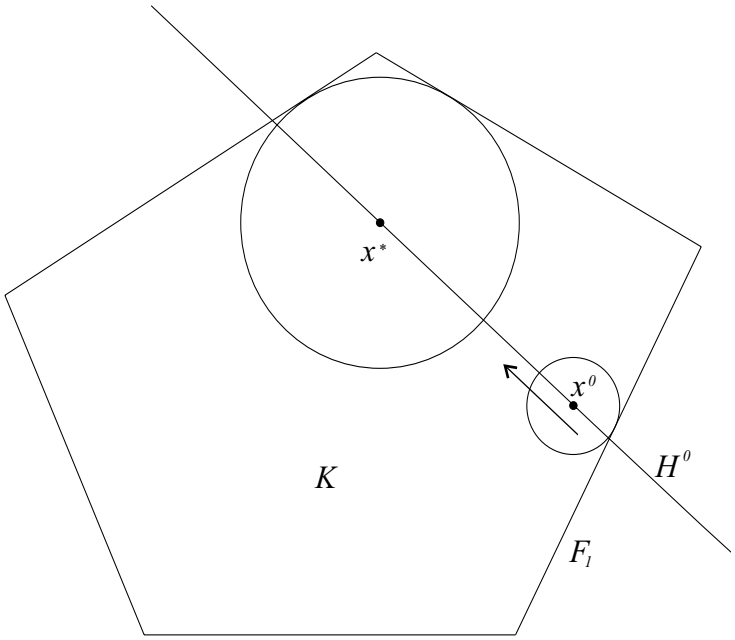
This procedure looks for profitable directions to move at current point $\hat{x}^s$ only among the set $\{P_{.1}, \ldots, P_{.m}\}$. If a profitable direction P in this set is found, it finds the optimum solution $(\bar{\theta}, \bar{\alpha})$ of (16) with this P, takes $\hat{x}^{s+1} = \hat{x}^s + \bar{\alpha}P$ if $\bar{\alpha}$ is finite, and continues the same way with $\hat{x}^{s+1}$ as the new point in the sequence. See Figure 8.

If $\bar{\alpha} = \infty$, then the objective value in the original LP (12) is unbounded below and its dual infeasible, and so the whole method terminates. This procedure stops when there are no profitable directions in the set $\{P_{.1}, \ldots, P_{.m}\}$, or when the improvement in the radius of the ball becomes small.

When there are several profitable directions to move at the current point $\hat{x}^s$ in the set $\{P_{.1}, \ldots, P_{.m}\}$ in this procedure, efficient selection criteria to choose the best among them can be developed. In fact, the best may be among the $P_{.i}$ that correspond to i that tie for the minimum in $\delta(\hat{x}^s) = \min\{A_{i.}\hat{x}^s - b_i\colon i = 1 \text{ to } m\}$, or a weighted average of these directions (even though this direction is not included in our list of directions to pursue).

As can be seen, the procedure used in this centering strategy does not need any matrix inversion, and only solves a series of two-variable LPs that can be solved very efficiently.

FIGURE 8. Moving the center from x^0 along the direction $P_{.1}$ to x^* leads to a larger ball inside K.



Procedure 2. Getting an Approximate Solution to the Centering Step. We noticed that at the beginning of solving this centering step, $T(\hat{x}^s)$ for small s has small cardinality and usually the set of row vectors $\{c, A_{i.}$ for $i \in T(\hat{x}^s)\}$ tends to be linearly independent. Whenever this set of row vectors is linearly independent, a profitable direction to move at $\hat{x}^s$ can be obtained by solving the following system of linear equations in P

$$\begin{aligned} A_{i.}P &= 1 \quad \text{for each } i \in T(\hat{x}^s) \\ cP &= 0. \end{aligned}$$

This system has a solution because the coefficient matrix has full row rank. Finding a solution to this system, of course, requires one matrix inversion operation. Using a solution P of this system as the profitable direction to move has the advantage that if the next point in the sequence is $\hat{x}^{s+1}$, then the corresponding set $T(\hat{x}^{s+1}) \supset T(\hat{x}^s)$. The same process can be continued if $\{c, A_{i.}$ for $i \in T(\hat{x}^{s+1})\}$ is again linearly independent. This process can be continued until we reach a point $\hat{x}^u$ for which $\{c, A_{i.}$ for $i \in T(\hat{x}^u)\}$ is linearly dependent. At that stage, this procedure shifts to Procedure 1 and continues as in Procedure 1.

Procedure 3. Getting an Approximate Solution to the Centering Step. Suppose the value of the objective function at the current interior feasible solution is t. Then the centering step at it is to

$$\text{maximize } \delta(x) = \min\{A_{i.}x - b_i\colon i = 1 \text{ to } m\} \text{ subject to } cx = t.$$

This is a nonsmooth optimization problem, efficient schemes for solving such max-min problems have been developed in nonsmooth convex minimization literature. One good example is Nestrov [35], which can be used to solve it. Also, the effectiveness of Procedure 1 can be improved by including in it some of the line-search directions used in these methods.

6.5.1. Numerical Example. We apply one iteration of this method on the fertilizer product mix problem (1) of §1.4 to illustrate the method, both numerically and with a figure.

We will use Procedure 1 for the centering step. Here is the problem in minimization form.

$$\begin{aligned} \text{Minimize} \quad & z = -15x_1 - 10x_2 \\ \text{subject to} \quad & 1{,}500 - 2x_1 - x_2 \geq 0 \\ & 1{,}200 - x_1 - x_2 \geq 0 \\ & 500 - x_1 \geq 0 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \end{aligned}$$

Normalizing the coefficient vectors of all the constraints and the objective function to Euclidean norm 1, here it is again.

$$\begin{aligned} \text{Minimize} \quad & z = -0.832x_1 - 0.555x_2 \\ \text{subject to} \quad & 670.820 - 0.894x_1 - 0.447x_2 \geq 0 \\ & 848.530 - 0.707x_1 - 0.707x_2 \geq 0 \\ & 500 - x_1 \geq 0 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \end{aligned} \tag{17}$$

6.6. The Centering Step

Let K denote the set of feasible solutions, and let $x^0 = (10, 1)^T$ be the initial interior feasible solution. When we plug in x^0 in the constraints in (17), the left side expressions have values 661.433, 840.753, 490, 10, 1, respectively. So, the radius of the largest ball inside K with x^0 as center is $\delta^0 = \min\{661.433, 840.753, 490, 10, 1\} = 1$.

The objective plane through x^0 is the straight line in R^2 defined by $-0.832x_1 - 0.555x_2 = -8.875$. This is the straight line joining $(10.667, 0)^T$ and $(0, 15.991)^T$ in the x_1, x_2-plane. So, the only direction on it is $P_{.1} = (10.667, -15.991)^T$. Moving from x^0 in the direction of $P_{.1}$, a step length α leads to the new point $(10 + 10.667\alpha, 1 - 15.991\alpha)^T$. Finding the optimum step length α leads to the following two-variable LP in variables θ and α (Table 6).

Because the minimum RHS constant in this problem occurs in only one row, from Result 1, we know that the optimum value of α in this problem will be nonzero. Actually, the optimum solution of this problem is $(\bar{\theta}, \bar{\alpha})^T = (6.4, -0.338)^T$. See Figure 9. The new position for the center is $\hat{x}^1 = x^0 - 0.338P_{.1} = (10, 1)^T - 0.338(10.667, -15.991)^T = (6.4, 6.4)^T$, and the maximum radius ball with it as center has radius 6.4. Because $P_{.1}$ is the only direction in $K \cap \{x\colon cx = cx^0\}$, in this case, this ball is the maximum radius ball inside K with center on the objective plane through x^0.

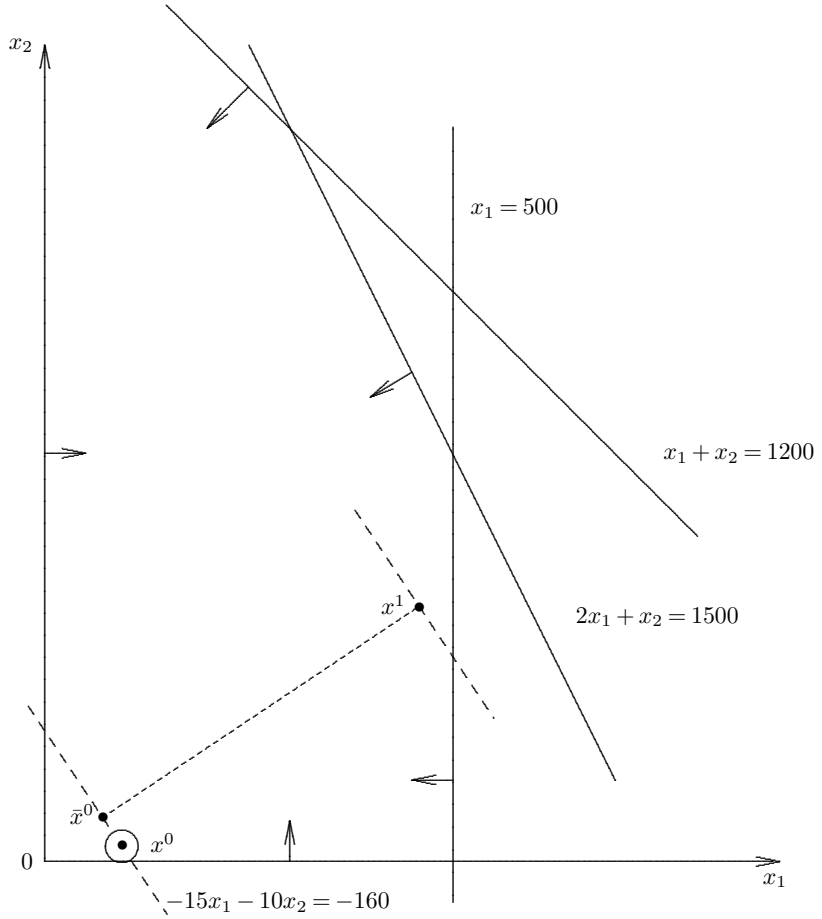
If we try to get a larger ball by moving from x^1 in the direction $P_{.1}$ a step length of α, it can be verified that in the two-variable LP to find the optimum step length α, the entries in

TABLE 6. The two variable LP in a line search step for centering.

θ	α		
1	2.388	$\leqq$	661.433
1	−3.765	$\leqq$	840.753
1	10.667	$\leqq$	490
1	−10.667	$\leqq$	10
1	15.991	$\leqq$	1
1	0		Maximize

$\theta \geqq 0$, α unrestricted.

FIGURE 9. This figure (not drawn to scale) shows feasible region K with five facets, each has an arrow pointing its feasible side.



Notes. Only a small sphere of radius 1 can be drawn inside K with initial point x^0 as center. Dashed line through x^0 is the objective plane, centering strategy moves point to $\bar{x}^0 = (6.4, 6.4)^T$ on this plane. With $\bar{x}^0$ as center, a sphere of radius 6.4 can be inscribed inside K. The descent move from $\bar{x}^0$ in Step 2 in direction $-c^T$ (dotted line) leads to $x^1 = (499, 335)^T$ with objective value $-10{,}835$. The dashed line through x^1 is the objective plane $\{x\colon -15x_1 - 10x_2 = -10{,}835\}$. Another iteration begins with x^1.

the RHS vector are 662.238, 839.48, 493.6, 6.4, 6.4, and the coefficient vector of α remains the same as in the above table. In this problem, the minimum RHS constant occurs in both Rows 4 and 5, and the coefficients of α in these two rows have opposite signs, indicating by Result 1 that the optimum value for step length α will be 0. This indicates that $\hat{x}^1$ is the best position for the center of the ball on the objective plane through x^0 in this problem, which in the algorithm is denoted by $\bar{x}^0$.

6.7. Descent Move Following Centering

The current center is $\bar{x}^0 = (6.4, 6.4)^T$. In this initial iteration, the only descent direction we have available at $\bar{x}^0$ is $-c^T = (0.832, 0.555)^T$. Moving from $\bar{x}^0$ a step length γ in the direction $-c^T$ leads to the point $(6.4 + 0.832\gamma, 6.4 + 0.555\gamma)^T$. Taking the tolerance $\epsilon = 1$, we see that the maximum step length is $\gamma = \min\{666.571, 854.72, 592.067\} = 592.067$. Fixing $\gamma = 592.067$, we get the new interior feasible solution $x^1 = (499, 335)^T$.

With x^1, we need to go to the next iteration and continue in the same way. Figure 9 illustrates both the centering step carried out beginning with the initial interior feasible solution x^0 and the descent move carried out here.

6.8. Some Advantages of This Method

Redundant constraints in a linear program can affect the efficiency for solving it by the simplex method, or the earlier interior point methods. In fact in Deza et al. [7], they show that when redundant constraints are added to the Klee-Minty problem over the n-dimensional cube, the central path in these methods takes $2^n - 2$ turns as it passes through the neighborhood of all the vertices of the cube before converging to the optimum solution.

Because gravitational methods and this method operate only with the touching constraints, their performance is not affected by redundant constraints. Also, redundant constraints in (12) do not correspond to facets of K. So, in the centering step, having redundant constraints in (12) just adds some additional directions $P_{.i}$ in the set of directions used in the centering Procedure 1. Programming tricks can be developed for efficiently selecting promising directions in this set to search for improving the value of $f(\alpha)$ in this procedure, and keep this centering procedure and this method efficient.

Also, because this method needs no matrix inversions when Procedure 1 is used for centering, it can be used even when A is dense.

6.9. Interpretation as a Predictor-Corrector Path-Following Interior Point Method

This method is a path-following interior point method that tries to follow the path of centers $\mathcal{P}$ defined above, just as the methods discussed in §4.5 try to follow the central path $\mathcal{C}$ defined there. This method is like the predictor-corrector path-following method PC discussed in §4.5. In each iteration of this method, Step 1 (the centering step) is like a corrector step—It tries to move the current interior feasible solution toward the path of centers $\mathcal{P}$ while keeping the objective value constant using line searches based on solving two-variable LP models if Procedure 1 is employed. Step 2 (the descent step) is like a predictor step moving the longest possible step in a descent direction.

The central path of §4.5 depends on the algebraic representation of the set of feasible solutions through the constraints in the problem being solved, and may become very long and crooked if there are many redundant constraints in the model. The path of centers $\mathcal{P}$ followed by this algorithm, however, is unaffected by redundant constraints in the model and only depends on the set of feasible solutions K of the problem as a geometric set.

6.10. Relation to the Geometric Method of Section 4.1

We will now show that this method can be viewed as computationally duplicating the geometric algorithm for solving two-variable LPs discussed in §4.1. In that method, the graph of the feasible region K is drawn on paper, a point $x^0 \in K$ is selected visually, and the straight line $z(x) = cx = cx^0$ (objective plane through x^0) is drawn. Looking at the picture of the feasible region, the objective plane is moved parallel to itself in the desirable direction as far as possible until any further move will make the line lose contact with the feasible region K. The intersection of K with the final position of the line is the set of optimum solutions of the LP.

Due to lack of visibility in higher-dimensional spaces to check whether the objective plane can be moved further in the desirable direction while still keeping its contact with the feasible region, this simple geometric method could not be generalized to dimensions $\geqq 3$. In this method, the centering step guarantees that in the descent step, the objective plane through the center $\bar{x}^r$ of the current ball $B(\bar{x}^r, \delta(\bar{x}^r))$ can move a distance of $\delta(\bar{x}^r)$ in the descent direction and still keep its contact with the feasible region. Thus, this method can be viewed as a generalization of the objective plane moving step in the geometric method for two dimensional LPs.

7. An Iterative Method for LP

The name *iterative method* usually refers to a method that generates a sequence of points using a simple formula for computing the $(r+1)$th point in the sequence as an explicit function of the rth point: like $\xi^{r+1} = f(\xi^r)$. An iterative method begins with an initial point ξ^0 (often chosen arbitrarily, or subject to simple constraints that are specified, such as $\xi^0 \geq 0$) and generates the sequence $\xi^0, \xi^1, \xi^2, \ldots$ using the above formula.

Their advantage is that they are extremely simple and easy to program (much more so than the methods discussed so far) and hence may be preferred for tackling very large problems lacking special structure. A variety of iterative methods have been developed for LP and shown to converge to an optimum solution in the limit under some assumptions. But so far these methods have not been popular because in practice the convergence rate has been observed to be very slow.

As an example, we discuss an iterative method known as the *saddle point algorithm* recently developed by Yi et al. [45] (see also Choi [4] and Kallio and Rosa [13]) that shows promise. They consider

the primal LP: minimixe $z = cx$, subject to $Ax = b$, $x \geq 0$
and the dual: maximize $b^T y$, subject to $A^T y \leq c^T$

where A is a matrix of order $m \times n$. The Lagrangian function for this primal-dual pair of LPs is $L(x,y) = cx - (Ax-b)^T y$ defined over $x \in R^n_+, y \in R^m$.

Starting with an arbitrary (x^0, y^0) satisfying $x^0 \geq 0$ and $y^0 \in R^m$, this algorithm generates a sequence of points (x^r, y^r), always satisfying $x^r \geq 0$, $r = 0, 1, \ldots$. For $r = 0, 1, \ldots$ we define corresponding to (x^r, y^r)

the dual slack vector $s^r = c^T - A^T y^r = \nabla_x L(x^r, y^r)$, and the primal constraint violation vector $v^r = b - Ax^r = \nabla_y L(x^r, y^r)$.

In (x^r, y^r) even though $x^r \geq 0$, v^r may be nonzero and s^r may not be nonnegative, so x^r may not be primal feasible and y^r may not be dual feasible.

The pair $(\bar{x}, \bar{y})$ is said to be a *saddle point* for this primal-dual pair of LPs if

$$L(\bar{x}, y) \leq L(\bar{x}, \bar{y}) \leq L(x, \bar{y}) \quad \text{for all } x \geq 0, \text{ and for all } y.$$

In LP theory, these conditions are called *saddle point optimality conditions*; if they are satisfied, $(\bar{x}, \bar{y})$ is called a *saddle point* for this primal-dual pair of LPs, and then $\bar{x}$ is an optimum solution for the primal and $\bar{y}$ is an optimum solution for the dual. The aim of this algorithm is to generate a sequence converging to a saddle point.

For any real number γ, define $\gamma^+ = \text{maximum}\{\gamma, 0\}$. For any vector $\xi = (\xi_j)$, define $\xi^+ = (\xi_j^+)$. We will now describe the general iteration in this algorithm. $\alpha > 0, \beta > 0$ are two step length parameters used in the iterative formula, typical values for them are: α (step-length parameter in the x-space), β (step-length parameter in the y-space), both equal to 10.

7.1. General Iteration $r+1$

Let (x^r, y^r) be the current point in the sequence. Compute $x^r_I = (x^r - \alpha s^r)^+$, $y^r_I = y^r + \beta v^r$, $\ell^r_x = L(x^r, y^r) - L(x^r_I, y^r)$, $\ell^r_y = L(x^r, y^r_I) - L(x^r, y^r)$, $\ell_r = \ell^r_x + \ell^r_y$.

It can be shown that ℓ^r_x, ℓ^r_y are both ≥ 0. If $\ell_r = 0$, then (x^r, y^r) is a saddle point, terminate the algorithm.

If $\ell_r > 0$, then compute $s^r_I = c^T - A^T y^r_I$, $v^r_I = b - Ax^r_I$, $\rho_r = \ell_r / (||s^r_I||^2 + ||v^r_I||^2)$, where $||\cdot||$ denotes the Euclidean norm. Let $x^{r+1} = (x^r + \rho_r s^r_I)^+$, $y^{r+1} = y^r + \rho_r v^r_I$. With (x^{r+1}, y^{r+1}) as the new current pair, go to the next iteration.

Under the assumption that both the primal and dual have feasible solutions, this algorithm has been proved to generate a sequence converging to a saddle point. In implementing this algorithm, instead of keeping the step-length parameters α, β fixed, their values can be chosen by line searches to optimize $L(x,y)$ (minimize with respect to x, maximize with respect to y).

8. Summary and Conclusion

We traced the history of mathematical models involving systems of linear constraints including linear inequalities, and linear programs; and algorithms for solving them. All existing methods in use for solving them need complicated matrix inversion operations, and are suitable for solving large-scale problems only when the data is very sparse. These methods encounter difficulties for solving large-scale dense problems, or even those that only have some important dense columns. We also discussed a new, efficient descent method that does not need matrix inversion operations and that shows great promise for solving large-scale problems fast.

References

[1] D. A. Bayer and J. C. Lagarias. The nonlinear geometry of linear programming, I. Affine and projective scaling trajectories, II. Legendre transform coordinates and central trajectories, III. Projective Legendre transform coordinates and Hilbert geometry. *Transactions of the American Mathematical Society* 314:499–581, 1989.

[2] S. Y. Chang. The steepest descent gravitational method for linear programming. Ph.D. dissertation, University of Michigan, Ann Arbor, MI, 1988.

[3] S. Y. Chang and K. G. Murty. The steepest descent gravitational method for linear programming. *Discrete Applied Mathematics* 25:211–239, 1989.

[4] B. Choi. Theory and algorithms for semidefinite programming. Ph.D. dissertation, University of Michigan, Ann Arbor, MI, 2001.

[5] G. B. Dantzig. *Linear Programming and Extensions.* Princeton University Press, Princeton, NJ, 1963.

[6] G. B. Dantzig and M. N. Thapa. *Linear Programming, 1. Introduction.* Springer-Verlag, New York, 1997.

[7] A. Deza, E. Nematollahi, R. Peyghami, and T. Terlaky. The central path visits all the vertices of the Klee-Minty cube. AdvOL-Report 2004/11, McMaster University, Hamilton, Ontario, Canada, 2004.

[8] I. I. Dikin. Iterative solution of problems of linear and quadratic programming. *Soviet Mathematics Doklady* 8:674–675, 1967.

[9] J. Farkas. Über die Anwendungen des mechanischen Princips von Fourier. *Mathematische und naturwissenschaftliche Berichte aus Ungarn* 12:263–281, 1895.

[10] D. Gale. *The Theory of Linear Economic Models.* McGraw-Hill, New York, 1960.

[11] P. Gordan. Ueber die Auflösung linearer Gleichungen mit reellen Coefficienten. *Mathematische Annalen* 6:23–28, 1873.

[12] O. Güler, C. Roos, T. Terlaky, and J.-P. Vial. A survey of the implications of the behavior of the central path for the duality theory of linear programming. *Management Science* 41:1922–1934, 1995.

[13] M. Kallio and C. Rosa. Large scale convex optimization via saddle point computation. *Operations Research* 47:373–395, 1999.

[14] S. Kangshen, John N. Crossley, and Anthony W. C. Lun. *9 Chapters on the Mathematical Art: Companion and Commentary.* Oxford University Press, Oxford, United Kingdom, and Science Press, Beijing, China, 1999.

[15] L. V. Kantorovich. *The Mathematical Method of Production Planning and Organization.* (In Russian, 1939). Transl. *Management Science* 6(4):363–422, 1960.

[16] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica* 4:373–395, 1984.

[17] M. Kojima, S. Mizuno, and A. Yoshise. A primal-dual interior point algorithm for linear programming. Ch. 2. N. Meggiddo, ed. *Progress in Mathematical Programming: Interior Point and Related Methods.* Springer-Verlag, New York, 29–47, 1989.

[18] V. Lakshmikantham and S. Leela. *The Origin of Mathematics.* University Press of America, Lanham, MD, 2000.

[19] L. McLinden. The analogue of Moreau's proximation theorem, with applications to the nonlinear complementarity problem. *Pacific Journal of Mathematics* 88:101–161, 1980.

[20] N. Meggiddo. Pathways to the optimal set in linear programming. Ch. 8. N. Meggiddo, ed. *Progress in Mathematical Programming: Interior Point and Related Methods.* Springer-Verlag, New York, 131–158, 1989.

[21] S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization* 2:575–601, 1992.

[22] H. Minkowski. *Geometrie der Zahlen* (*Erste Lieferung*). Teubner, Leipzig, Germany, 1896.

[23] S. Mizuno, M. Todd, and Y. Ye. On adaptive step primal-dual interior point algorithms for linear programming. *Mathematics of Operations Research* 18:964–981, 1993.

[24] R. D. C. Monteiro and I. Adler. Interior path-following primal-dual algorithms, Part I: Linear programming. *Mathematical Programming* 44:27–41, 1989.

[25] T. L. Morin, N. Prabhu, and Z. Zhang. Complexity of the gravitational method for linear programming. *Journal of Optimization Theory and Applications* 108:633–658, 2001.

[26] K. G. Murty. *Linear Programming.* Wiley, New York, 1983.

[27] K. G. Murty. The gravitational method for linear programming. *Opsearch* 23:206–214, 1986.

[28] K. G. Murty. *Linear Complementarity, Linear and Nonlinear Programming.* Helderman Verlag, Berlin, Germany, 1988.

[29] K. G. Murty. *Computational and Algorithmic Linear Algebra and n-dimensional Geometry.* http://ioe.engin.umich.edu/people/fac/books/murty/algorithmic_linear_algebra/, 2004.

[30] K. G. Murty. A gravitational interior point method for LP. *Opsearch* 42(1):28–36, 2005.

[31] K. G. Murty. *Optimization Models for Decision Making*, Vol. 1. http://ioe.engin.umich.edu/people/fac/books/murty/opti_model/, 2005.

[32] K. G. Murty. My experiences with George Dantzig. http://www.informs.org/History/dantzig/rem_murty.htm, 2005.

[33] K. G. Murty. A new practically efficient interior point method for LP. *Algorithmic Operations Research* 1:3–19.

[34] K. G. Murty and Y. Fathi. A critical index algorithm for nearest point problems on simplicial cones. *Mathematical Programming* 23:206–215, 1982.

[35] Y. Nestrov. Smooth minimization of non-smooth functions. *Mathematical Programming Series A* 103:127–152, 2005.

[36] R. Saigal. *Linear Programming: A Modern Integrated Analysis.* Kluwer Academic Publishers, Boston, MA, 1995.

[37] A. Schrijver. *Theory of Linear and Integer Programming.* Wiley-Interscience, New York, 1986.

[38] G. Sonnevend, J. Stoer, and G. Zhao. On the complexity of following the central path of linear programming by linear extrapolation. *Mathematics of Operations Research* 62:19–31, 1989.

[39] J. Von Neumann. Discussion of a maximum problem. A. H. Taub, ed., *John von Neumann, Collected Works*, Vol VI. Pergamon Press, Oxford, England, 89–95, 1963.

[40] D. R. Wilhelmsen. A nearest point algorithm for convex polyhedral cones and applications to positive linear approximation. *Mathematics of Computation* 30:48–57, 1976.

[41] P. Wolfe. Algorithm for a least distance programming problem. *Mathematical Programming Study 1* 190–205, 1974.

[42] P. Wolfe. Finding the nearest point in a polytope. *Mathematical Programming* 11:128–149, 1976.

[43] S. J. Wright. *Primal-Dual Interior-Point Methods.* SIAM, Philadelphia, PA, 1997.

[44] Y. Ye. *Interior Point Algorithms, Theory and Analysis.* Wiley-Interscience, New York, 1997.

[45] S. Yi, B. Choi, R. Saigal, W. Zhu, and M. Troutt. Convergence of a gradient based algorithm for linear programming that computes a saddle point. Technical report, University of Michigan, Ann Arbor, MI, 1999.

TUTORIALS IN
OPERATIONS RESEARCH
INFORMS 2006

 | ISBN 13 978-1-877640-20-4
DOI 10.1287/educ.1063.0029

Semidefinite and Second-Order Cone Programming and Their Application to Shape-Constrained Regression and Density Estimation

Farid Alizadeh
Department of Management Science and Information Systems and Rutgers Center for Operations Research, Rutgers, the State University of New Jersey, 640 Bartholomew Road, Piscataway, New Jersey 08854, alizadeh@rutcor.rutgers.edu

Abstract In statistical analysis often one wishes to approximate a functional relationship between one or more explanatory variables and one or more response variables, with the additional condition that the resulting function satisfy certain "shape constraints." For instance, we may require that our function be nonnegative, monotonic, convex, or concave. Such problems arise in many areas from econometrics to biology to information technology. It turns out that often such shape constraints can be expressed in the form of semidefinite constraints on certain matrices. Therefore, there is an intimate connection between shape-constrained regression or approximation and the optimization problems known as semidefinite programming. In this tutorial, we first present a broad introduction to the subject of semidefinite programming and the related problem of second-order cone programming. We review duality theory complementarity and interior point algorithms. Next, we survey some properties of nonnegative polynomials and nonnegative spline functions of one or possibly several variables that can be expressed as sum of squares of other functions. On the one hand, these classes of functions are characterized by positive semidefinite matrices. On the other hand, they are excellent choices for approximating unknown functions with high precision. Finally, we review some concrete problems arising in parametric and nonparametric regression and density estimation problems with additional nonnegativity or other shape constraints that can be approached by nonnegative polynomials and splines, and can be solved using semidefinite programming.

Keywords semidefinite programming; second-order cone programming; nonparametric density estimation; nonparametric shape-constrained regression

1. Introduction and Background

Semidefinite programming (SDP) is a field in optimization theory that unifies several classes of convex optimization problems. In most cases, the feasible set of the problem is expressed either as matrix valued functionals that are required to be positive semidefinite, or they are positive semidefinite matrices that are required to satisfy additional linear constraints. First, recall that a symmetric matrix A is positive semidefinite (respectively, positive definite) if any of the following equivalent statements hold:

(1) For all vectors $\mathbf{x}$, $\mathbf{x}^\top A\mathbf{x} \geq 0$ (respectively, for all $\mathbf{x} \neq 0$, $\mathbf{x}^\top A\mathbf{x} > 0$),

(2) All eigenvalues of A are nonnegative (respectively, all eigenvalues of A are positive), (Recall that all eigenvalues of a symmetric matrix are always real numbers).

(3) There is a matrix B such that $A = B^\top B$, where B is any arbitrary matrix (respectively, there is a full-rank matrix B such that $B^\top B = A$). The matrix B need not even be a square matrix.

Positive definite matrices are nonsingular positive semidefinite matrices.

For two symmetric matrices A and B we write $A \succcurlyeq B$ (respectively, $A \succ B$) if $A - B$ is positive semidefinite (respectively, positive definite); in particular, $A \succcurlyeq 0$ means A is positive semidefinite. A particular version of (3) can be stated as follows.

Lemma 1.1. *For every positive semidefinite* (*respectively, positive definite*) *matrix X there is a unique positive semidefinite* (*respectively, positive definite*) *matrix Y such that $Y^2 = X$. We write $X^{1/2}$ for Y.*

It is well known and easy to see that the set of all positive semidefinite matrices is a *convex cone*: If $A \succcurlyeq 0$, then $\alpha A \succcurlyeq 0$ for all $\alpha \geq 0$, and if $A \succcurlyeq 0, B \succcurlyeq 0$, then $A + B \succcurlyeq 0$ (simply apply (1)). This cone is closed, its interior is the set of all positive definite matrices, and its boundary consists of singular positive semidefinite matrices.

Now, semidefinite programs are optimization problems that may have any number of constraints of the form

$$\text{(a)} \ \sum_i x_i A_i \succcurlyeq A_0 \quad \text{or} \quad \text{(b)} \ X \succcurlyeq 0, \ A_i \bullet X = b_i$$

where decision variables in (a) are the x_i and in (b) are the individual entries X_{ij} of the symmetric matrix X. Also, $X \bullet Y = \sum_{ij} X_{ij} Y_{ij}$ is the inner product of matrices X and Y. There are many classes of optimization problems that can be expressed as semidefinite programs. Examples arise from combinatorial optimization, statistics, control theory, finance, and various areas of engineering, among others. In this paper, we will focus on a particular set of applications in statistics and approximation theory (see §4 below). However, let us briefly mention a number of ways that semidefinite programs arise in other contexts.

One common way semidefinite programs arise in applications is through minimizing (or maximizing) certain functions of eigenvalues of symmetric matrices. Let $\lambda_1(A) \geq \lambda_2(A) \geq \cdots \geq \lambda_n(A)$ be largest to smallest eigenvalues of a symmetric matrix A. Also, let $\lambda^{(k)}(A)$ be the kth largest eigenvalue of A absolute valuewise: $|\lambda^{(1)}(A)| \geq \cdots \geq |\lambda^{(n)}(A)|$. Similarly for an arbitrary $m \times n$ matrix B, let $\sigma_k(B)$ be the kth largest singular value of B. Then, a number of optimization problems involving eigenvalues can be expressed as semidefinite programs. For example, consider the following problem:

$$\min_{\mathbf{x}} \lambda_1 \left(A_0 + \sum_i x_i A_i \right). \tag{1}$$

The standard way to to express this problem is to create a new variable z and express (1) as

$$\begin{aligned} \min \quad & z \\ \text{s.t.} \quad & zI - \sum_i x_i A_i \succcurlyeq A_0, \end{aligned} \tag{2}$$

which is a semidefinite program with a linear objective function. More generally, the following extensions can be expressed as semidefinite programs. Let $A(\mathbf{x}) = \sum_i x_i A_i$ for symmetric matrices A_i, and let $B(\mathbf{x}) = \sum_i x_i B_i$ for arbitrary matrices B_i all, say $m \times n$.

(1) Maximize the smallest eigenvalue of $A(\mathbf{x})$: $\max_{\mathbf{x}} \lambda_n(A(\mathbf{x}))$.
(2) Minimize the absolute-valuewise largest eigenvalue of $A(\mathbf{x})$: $\min_{\mathbf{x}} |\lambda^{(1)}(A(\mathbf{x}))|$.
(3) Minimize the largest singular value of $B(\mathbf{x})$: $\min_{\mathbf{x}} \sigma_1(B(\mathbf{x}))$.
(4) Minimize sum of the k largest eigenvalues of $A(\mathbf{x})$: $\min_{\mathbf{x}} \sum_{i=1}^{k} \lambda_i(A(\mathbf{x}))$.
(5) Maximize sum of the k smallest eigenvalues of $A(\mathbf{x})$: $\max_{\mathbf{x}} \sum_{i=1}^{k} \lambda_{n-i}(A(\mathbf{x}))$.
(6) Minimize sum of the k absolute-valuewise largest eigenvalues of
$$A(\mathbf{x})\text{: } \min_{\mathbf{x}} \sum_{i=1}^{k} |\lambda^{(i)}(A(\mathbf{x}))|.$$
(7) Minimize sum of the k largest singular values of $B(\mathbf{x})$: $\min_{\mathbf{x}} \sum_{i=1}^{k} \sigma_i(B(\mathbf{x}))$.

(8) Minimize a particular weighted sum of the k largest eigenvalues of $A(A\mathbf{x})$:
$\min_{\mathbf{x}} \sum_{i=1}^{k} w_i \lambda_i(A(\mathbf{x}))$ and $\sum_{i=1}^{k} w_i |\lambda^{(i)} A(\mathbf{x}))|$ for $w_1 \geq w_2 \geq \cdots \geq w_k > 0$.
(9) Minimize a particular weighted sum of k largest singular values of $B(\mathbf{x})$:
$\min_{\mathbf{x}} \sum_{i}^{k} w_i \sigma_i(B(\mathbf{x}))$ SDP.

Other SDP representations that are based on the simple inequality $z \leq \sqrt{xy}$ (where $x, y, z \geq 0$), which is equivalent to $z^2 \leq xy$ that in turn is equivalent to 2×2 semidefinite constraint:

$$\begin{pmatrix} x & z \\ z & y \end{pmatrix} \succcurlyeq 0.$$

This equivalence is quite simple but can be iterated to express quite complicated inequalities. The following problem should shed light on how to accomplish this in a more general setting. Consider

$$\begin{aligned} \max \quad & x_1 x_2 \cdots x_n \\ \text{s.t.} \quad & A\mathbf{x} = \mathbf{c} \\ & \mathbf{0} \leq \mathbf{a} \leq \mathbf{x} \leq \mathbf{b} \end{aligned} \tag{3}$$

where $\mathbf{x} = (x_1, \ldots, x_n)$. Now we can replace the objective function with $(x_1 \ldots x_n)^{1/n}$ without changing the problem. Write

$$(x_1 \ldots x_n)^{1/n} = \sqrt{(x_1 \cdots x_{n/2})^{2/n} (x_{1+n/2} \cdots x_n)^{2/n}}.$$

(3) can now be written as

$$\begin{aligned} \max \quad & z \\ \text{s.t.} \quad & z \geq \sqrt{z_1 z_2} \\ & z_1 \geq (x_1 \cdots x_{n/2})^{2/n} \\ & z_2 \geq (x_{1+n/2} \cdots x_n)^{2/n} \\ & A\mathbf{x} = \mathbf{c}, \quad \mathbf{0} \leq \mathbf{a} \leq \mathbf{x} \leq \mathbf{b}. \end{aligned} \tag{4}$$

Applying recursively the same trick to z_1 and z_2, we turn (4) to a semidefinite program with n 2×2 semidefinite constraints. In this case, the problem can be represented by simpler second-order cone programming (SOCP) constraints; we will develop this concept more fully in the section to follow. Many more examples of SDP are given in Alizadeh [1], Nesterov and Nemirovski [13], and Vandenberghe and Boyd [21]. Also, the papers collected in Saigal et al. [17] contain many other problems that can be modeled as SDP.

1.1. Second-Order Cone Programming (SOCP)

A problem that is closely related to SDP is the SOCP. A simple second-order cone is defined as follows: Let $\mathbf{x} = (x_0, x, \ldots, x_n)$, thus, $\mathbf{x}$ is indexed from zero, and write $\bar{\mathbf{x}} = (x_1, x_2, \ldots, x_n)$. Then, the second-order cone is

$$\mathcal{Q}_{n+1} = \{\mathbf{x} \mid x_0 \geq \|\bar{\mathbf{x}}\|\}$$

where $\|\bar{\mathbf{x}}\|$ is the euclidean norm of $\bar{\mathbf{x}}$. Thus, the condition for membership in second-order cone programs is $x_0 \geq \left(x_1^2 + \cdots + x_n^2\right)^{1/2}$.

A general second-order cone is composed of multiple vectors of possibly different sizes, each of which belongs to a simple second-order cone:

$$\mathcal{Q} = \{(\mathbf{x}_1, \ldots, \mathbf{x}_m) \mid \mathbf{x}_i \in \mathcal{Q}_{i+1}, \text{ for } i = 1, \ldots, m\}.$$

The interior of the second-order cone consists of all vectors $\mathbf{x}$ where $x_0 > \|\bar{\mathbf{x}}\|$, and its boundary consists of vectors where $x_0 = \|\bar{\mathbf{x}}\|$.

A second-order cone inequality (an SOC inequality) written as $\mathbf{x} \succcurlyeq_{\mathcal{Q}} \mathbf{y}$ (respectively, $\mathbf{x} \succ_{\mathcal{Q}} \mathbf{y}$) means that $\mathbf{x} - \mathbf{y} \in \mathcal{Q}$ (respectively, $\mathbf{x} - \mathbf{y} \in \operatorname{Int} \mathcal{Q}$).

A second-order cone optimization problem involves inequalities of the form

$$\sum_i x_i \mathbf{v}_i \succcurlyeq_{\mathcal{Q}} \mathbf{v}_0 \quad \text{or} \quad A\mathbf{x} = \mathbf{b},\ \mathbf{x} \succcurlyeq_{\mathcal{Q}} 0.$$

As in SDP, many optimization problems can be formulated as SOCP. In fact, inequalities of the form $z^2 \leq \mathbf{x}^\top \mathbf{y}$ can be reformulated as SOC inequalities as follows:

$$z^2 \leq \mathbf{x}^\top \mathbf{y} = \left\| \frac{\mathbf{x}+\mathbf{y}}{2} \right\|^2 - \left\| \frac{\mathbf{x}-\mathbf{y}}{2} \right\|^2.$$

Therefore,

$$\begin{pmatrix} \mathbf{x}+\mathbf{y} \\ \mathbf{x}-\mathbf{y} \\ z \end{pmatrix} \succcurlyeq_{\mathcal{Q}} \mathbf{0}.$$

Indeed, this transformation includes inequalities of the form $z \geq \sqrt{xy}$, and thus problems in (4) are in fact instances of SOCP.

As a special case, consider convex quadratic inequalities of the form

$$(\mathbf{x} - \mathbf{a})^\top Q(\mathbf{x} - \mathbf{a}) \leq b \tag{5}$$

where the matrix $Q \succcurlyeq 0$. In that case, there is a matrix A such that $Q = A^\top A$. Now, write (5) as

$$(\mathbf{x} - \mathbf{a})^\top A^\top A(\mathbf{x} - \mathbf{a}) \leq b$$

We see that it is of the form $\mathbf{y}^\top \mathbf{y} \leq z^2$ for $\mathbf{y} = A(\mathbf{x} - \mathbf{a})$, because b is necessarily positive. Constraints of the form (5) arise quite often in applications. One interesting class of examples are in portfolio optimization using Markowitz-type risk/return relations. Alizadeh and Goldfarb [2] and Lobo et al. [11] present many more examples of SOCP.

2. Cone-LP Framework, Duality, and Complementarity

In this section, we establish optimality conditions and duality theory for semidefinite and second-order cone-constrained problems, then extend these properties to more general optimization problems.

2.1. Duality and Complementary for Semidefinite Programming

Let us first start with the case where the objective function is linear. In the SDP problem, we can transform problems into the following standard format that we call the *primal*:

$$\begin{aligned} \min \quad & C_1 \bullet X_1 + \cdots + C_n \bullet X_n \\ \text{s.t.} \quad & \sum_{j=1}^{n} A_{ij} \bullet X_j = b_i \quad \text{for } i = 1, \ldots m \\ & X_i \succcurlyeq 0 \quad \text{for } i = 1, \ldots, n. \end{aligned} \tag{6}$$

Here each X_i is an $n_i \times n_i$ symmetric matrix. Note that when all $n_i = 1$, then the problem reduces to linear programming.

Associated with each semidefinite program there is another one that we call its *dual*. The dual of (6) is

$$\begin{aligned} \max \quad & \mathbf{b}^\top \mathbf{y} \\ \text{s.t.} \quad & \sum_{i=1}^{m} y_i A_{ij} + S_j = C_j \quad \text{for } j = 1, \cdots m \\ & S_j \succcurlyeq 0. \end{aligned} \tag{7}$$

It turns out under some general conditions, the optimal values of primal and dual problems coincide. In fact, if X_i's are feasible for the primal and $\mathbf{y}$ and S_i are feasible for the dual, then

$$\begin{aligned} \sum_i b_i y_i - \sum_j C_j \bullet X_j &= \sum_i \left(\sum_j A_{ij} \bullet X_j \right) y_i - \sum_j C_j \bullet X_j \\ &= \sum_j \left(C_j - \sum_i A_{ij} y_i \right) \bullet X_j \\ &= \sum_j S_j \bullet X_j \geq 0. \end{aligned}$$

The last inequality follows from the fact that if $X, S \succcurlyeq 0$, then $X \bullet S \geq 0$.

Thus, if we have X_j primal feasible, and $\mathbf{y}$ and S_j dual feasible, and $\mathbf{b}^\top \mathbf{y} - \sum_j C_j \bullet X_j = 0$, then X_j's are optimal for the primal, and $\mathbf{y}$ and S_j's are optimal for the dual. This fact is often referred as the *weak duality theorem*. The key question is whether the converse is true. That is, if the primal and the dual are both feasible, do the optimal values for each coincide? Unlike the case of linear programming—in which this is always true—it can be shown that in SDP, there are pathological cases in which the primal and dual optimal values are unequal. However, if there are strictly positive definite matrices X_j feasible for the primal or strictly positive definite matrices S_j feasible for the dual, then the values of objective functions for the primal and dual will be equal. This fact is known as *strong duality theorem* and plays a fundamental role in design of algorithms. We summarize this in the following theorem.

Theorem 2.1. Strong Duality for Semidefinite Programming. *Assume at least one of the following statements is true:*

- *There are symmetric positive definite matrices $X_1, \ldots, X_n$ feasible for the primal problem.*
- *There is a vector $\mathbf{y}$ and symmetric positive definite matrices $S_1, \ldots, S_n$ feasible for the dual problem.*

Then,

i. *If the primal problem is unbounded, that is, there is a sequence of feasible matrices $X_1^{(k)}, \ldots, X_n^{(k)}$ such that the value of the objective function $z_k = \sum_i C_i \bullet X_i^{(k)}$ tends to infinity as $k \to \infty$, then the dual problem is infeasible.*

ii. *If the dual problem is unbounded, that is, there is a sequence of feasible vectors $\mathbf{y}^{(k)}$ and matrices $S_i^{(k)}$ such that the objective function $u_k = \mathbf{b}^\top \mathbf{y}^{(k)}$ tends to infinity as $k \to \infty$, then the primal problem is infeasible.*

iii. *If both primal and dual problems are feasible and bounded, then for the optimal primal solution $X_1^*, \ldots, X_n^*$ and the optimal dual solution $\mathbf{y}^*$ and $S_1^*, \ldots, S_n^*$ we have*

$$C_1 \bullet X_1^* + \cdots + C_n \bullet X_n^* = \mathbf{b}^\top \mathbf{y}^* \quad \textit{equivalently} \quad X_i^* \bullet S_i^* = 0 \quad \textit{for } i = 1, \ldots, n.$$

Strong duality leads to a fundamental property, called *complementary slackness theorem*. We saw earlier that for primal and dual feasible $X_1, \ldots, X_n, \mathbf{y}, S_1, \ldots, S_n$, the size of the

duality gap equals $\sum_i X_i \bullet S_i$. To see how complementarity works, let us first state a simple lemma:

Lemma 2.1. *If X and Y are positive semidefinite matrices and $X \bullet Y = 0$, then $XY = 0$ and equivalently $XY + YX = 0$.*

To see this, first observe that $A \bullet B = B \bullet A$. Thus,

$$0 = X \bullet Y = \text{Trace}(XY) = \text{Trace}(XY^{1/2}Y^{1/2}) = (XY^{1/2}) \bullet Y^{1/2}$$
$$= Y^{1/2} \bullet (XY^{1/2}) = \text{Trace}(Y^{1/2}XY^{1/2}) \geq 0.$$

The last inequality comes from the fact that $Y^{1/2}XY^{1/2}$ is positive semidefinite and all of its eigenvalues are nonnegative, and therefore so is its trace. Now if $\text{Trace}(Y^{1/2}XY^{1/2}) = 0$, then sum of its nonnegative eigenvalues is zero; thus, each of the eigenvalues must be zero. However if all of the eigenvalues of $Y^{1/2}XY^{1/2}$ are zero, then all of the eigenvalues of XY are zero because XY and $Y^{1/2}XY^{1/2}$ have the same eigenvalues. This implies that $XY = 0$. By symmetry, $YX = 0$ and thus $XY + YX = 0$. The converse is obvious: If $YX = 0$, then $\text{Trace}(XY) = 0$. It takes a little bit of algebraic manipulation to show that if $XY + YX = 0$ and $X, Y \succcurlyeq 0$, then $XY = 0$; we omit this derivation here.

Now at the optimal value of primal and dual SDP problems, where the duality gap is zero, we have $0 = \sum_i X_i \bullet S_i$. Because each of $X_i \bullet S_i$ are nonnegative and they add up to zero, each of them must be zero. However, $X_i^* \bullet S_i^* = 0$ implies that $X_i^* S_i^* + S_i^* X_i^* = 0$. This is the complementarity slackness theorem for SDP.

Theorem 2.2. Complementarity Slackness for SDP. *If X_i^* and $\mathbf{y}^*, S_i^*$ are optimal solutions for primal and dual semidefinite programs and strong duality holds, then $X_i^* S_i^* + S_i^* X_i^* = 0$ for $i = 1, \ldots, n$.*

There are two important implications of the complementary slackness theorem. First, we can identify whether given primal and dual feasible solutions are optimal. Second, we can design algorithms in which a sequence of primal and dual solutions $X(k)$, $\mathbf{y}(k)$, and $S(k)$ converge toward feasibility and zero duality gap simultaneously. We will discuss a class of such problems below in §5.

2.1.1. Lagrange Multipliers for SDP with Nonlinear Objective. In many applications, we may have a problem in which the constraints are as in (6) or (7), but the objective function may be a general convex (or concave for the maximization problem) function. Let us assume $g(\mathbf{y})$ is a function that is at least twice-differentiable and concave. Consider the dual problem (7) with the objective replaced by a concave function $g(\mathbf{y})$. To make the presentation simple, we assume only one set of semidefinite inequalities.

$$\begin{aligned} \max \quad & g(\mathbf{y}) \\ \text{s.t.} \quad & C - \sum_i y_i A_i \succcurlyeq 0 \end{aligned} \tag{8}$$

Here, the constraint involves $n \times n$ matrices. Associating a Lagrange multiplier matrix X to the inequality in (8), the Lagrangian can be defined as

$$L(\mathbf{y}, X) = g(\mathbf{y}) + X \bullet \left(C - \sum_i y_i A_i \right). \tag{9}$$

Now the first-order optimality conditions can be stated as follows.

Theorem 2.3. *Assume that there exists y_i such that $C - \sum_i y_i A_i \succ 0$. Then a necessary condition for a feasible vector $\mathbf{y}^*$ to be an optimal solution of (8) is that there exists a symmetric matrix X where the following relations hold:*

$$\nabla_{\mathbf{y}} L = \nabla_{\mathbf{y}} g(\mathbf{y}) - (X \bullet A_1, \ldots, X \bullet A_m) = 0 \tag{10}$$

$$X\left(C - \sum_i y_i A_i\right) + \left(C - \sum_i y_i A_i\right) X = 0 \tag{11}$$

$$X \succcurlyeq 0, \tag{12}$$

where $\nabla_{\mathbf{y}} g(\cdot)$ is the gradient of $g(\mathbf{y})$.

2.2. Duality and Complementarity for Second-Order Cones

Similar to SDP, we can define a standard form for SOCP problems. Define the *primal* SOCP problem as

$$\begin{aligned} \min \quad & \mathbf{c}_1^\top \mathbf{x}_1 + \cdots + \mathbf{c}_n^\top \mathbf{x}_n \\ \text{s.t.} \quad & A_1 \mathbf{x}_1 + \cdots + A_n \mathbf{x}_n = \mathbf{b} \\ & \mathbf{x}_i \succcurlyeq_{\mathcal{Q}} 0 \quad \text{for } i = 1, \ldots, n. \end{aligned} \tag{13}$$

Let us define an associated dual problem:

$$\begin{aligned} \max \quad & \mathbf{b}^\top \mathbf{y} \\ \text{s.t.} \quad & A_i^\top \mathbf{y} + \mathbf{s}_i = \mathbf{c}_i \quad \text{for } i = 1, \ldots, n \\ & \mathbf{s}_i \succcurlyeq_{\mathcal{Q}} 0 \quad \text{for } i = 1, \ldots, n. \end{aligned} \tag{14}$$

Duality theorem results for SOCP may be stated in a form similar to those for SDP. First, if $\mathbf{x} = (x_0, \bar{\mathbf{x}}) \in \mathcal{Q}$, and $\mathbf{y} = (y_0, \bar{\mathbf{y}}) \in \mathcal{Q}$, then

$$\mathbf{x}^\top \mathbf{y} = x_0 y_0 + \bar{\mathbf{x}}^\top \bar{\mathbf{y}} \geq \|\bar{\mathbf{x}}\| \; \|\bar{\mathbf{y}}\| + \bar{\mathbf{x}}^\top \bar{\mathbf{y}} \geq |\bar{\mathbf{x}}^\top \bar{\mathbf{y}}| + \bar{\mathbf{x}}^\top \bar{\mathbf{y}} \geq 0.$$

This fact leads to the *weak duality theorem*: If $\mathbf{x}_i$ are primal feasible,

$$\begin{aligned} \sum_i \mathbf{c}_i^\top \mathbf{x}_i - \mathbf{b}^\top \mathbf{y} &= \sum_i \mathbf{c}_i^\top \mathbf{x}_i - \left(\sum_i A_i \mathbf{x}_i\right)^\top \mathbf{y} \\ &= \sum (\mathbf{c}_i^\top - \mathbf{y}^\top A_i)^\top \mathbf{x}_i \\ &= \sum \mathbf{x}_i \mathbf{s}_i \geq 0. \end{aligned}$$

The strong duality theorem for SDP can be developed similarly.

Theorem 2.4. Strong Duality for Second-Order Cone Programming. *Assume at least one of the following statements is true:*

- *There are primal feasible vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ where $\mathbf{x}_{i0} > \|\bar{\mathbf{x}}_i\|$ for all $i = 1, \ldots, n$.*
- *There are dual feasible vectors $\mathbf{y}$ and $\mathbf{s}_1, \ldots, \mathbf{s}_n$, such that $\mathbf{s}_{i0} > \|\bar{\mathbf{s}}_i\|$ for all $i = 1, \ldots, n$.*

Then,

i. *If the primal problem is unbounded, that is, there is a sequence of feasible vectors $\mathbf{x}_1^{(k)}, \ldots, \mathbf{x}_n^{(k)}$, such that the value of the objective function $z_k = \sum_i \mathbf{c}_i^\top \mathbf{x}_i^{(k)}$ tends to infinity as $k \to \infty$, then the dual problem is infeasible.*

ii. *If the dual problem is unbounded, that is, there is a sequence of feasible vectors $\mathbf{y}^{(k)}$ and vectors $\mathbf{s}_i^{(k)}$, such that the objective function $u_k = \mathbf{b}^\top \mathbf{y}^{(k)}$ tends to infinity as $k \to \infty$, then the primal problem is infeasible.*

iii. *If both primal and dual problems are feasible and bounded, then for the optimal primal solution* $(\mathbf{x}_1^*, \ldots, \mathbf{x}_n^*)$ *and the optimal dual solution* $\mathbf{y}^*$ *and* $(\mathbf{s}_1^*, \ldots, \mathbf{s}_n^*)$*, we have*

$$\mathbf{c}_1^\top \mathbf{x}_1^* + \cdots + \mathbf{c}_n \mathbf{x}_n^* = \mathbf{b}^\top \mathbf{y}^* \qquad \textit{equivalently} \qquad (\mathbf{x}_i^*)^\top \mathbf{s}_i^* = 0 \quad \textit{for } i = 1, \ldots, n.$$

The strong duality theorem for SOCP leads to the complementary slackness theorem. Again, we first develop an important lemma.

Suppose, $\mathbf{x}^\top \mathbf{y} = 0$ and $\mathbf{x}, \mathbf{y} \in \mathcal{Q}$. For now, assume that $x_0 \neq 0$ and $y_0 \neq 0$. Write

$$x_0 y_0 = -x_1 y_1 - \cdots - x_n y_n, \tag{15}$$

which can be written as

$$2x_0^2 = -2x_1 \frac{x_0}{y_0} - \cdots - 2x_n y_n \frac{x_0}{y_0}. \tag{16}$$

Next, write

$$y_0^2 \geq y_1^2 + \cdots + y_n^2 \tag{17}$$

or, equivalently,

$$x_0^2 \geq y_1^2 \frac{x_0^2}{y_0^2} + \cdots + y_n^2 \frac{x_0^2}{y_0^2}, \tag{18}$$

and finally

$$x_0^2 \geq x_1^2 + \cdots + x_n^2. \tag{19}$$

Summing the two sides of (16), (18), (19), we get

$$0 \geq \left(x_1^2 + y_1^2 \frac{x_0^2}{y_0^2} - 2x_1 \frac{x_0}{y_0} \right) + \cdots + \left(x_n^2 + y_n^2 \frac{x_0^2}{y_0^2} - 2x_n \frac{x_0}{y_0} \right) \tag{20}$$

$$= \sum_i \left(x_i + y_i \frac{x_0}{y_0} \right)^2. \tag{21}$$

Because the sum of a number of square numbers cannot add up to zero unless each one equals to zero, we get

Lemma 2.2. *If* $\mathbf{x}, \mathbf{y} \in \mathcal{Q}$ *and* $\mathbf{x}^\top \mathbf{y} = 0$*, then*

$$x_0 y_i + y_0 x_i = 0 \quad \textit{for } i = 1, \ldots n. \tag{22}$$

When $x_0 = 0$ (respectively, $y_0 = 0$), then, necessarily $\mathbf{x} = 0$ (respectively, $\mathbf{y} = 0$), and the lemma is obviously true.

Now if $\mathbf{x}_i^*$, $\mathbf{y}^*$, and $\mathbf{s}_i^*$ are primal and dual optimal and the strong optimality theorem holds, then at the optimum, the duality gap $0 = \sum_i \mathbf{c}_i^\top \mathbf{x}_i^* - \mathbf{b}^\top \mathbf{y}^* = \sum_i \left(\mathbf{x}_i^*\right)^\top \mathbf{s}_i^\top$. Thus, we get the complementary slackness theorem for SOCP.

Theorem 2.5. Complementary Slackness for SOCP. *If* $\mathbf{x}_i^*, \mathbf{y}^*, \mathbf{s}_i^*$ *are optimal solutions for primal and dual semidefinite programs and strong duality holds, then*

$$\mathbf{x}_{i0}^* \mathbf{s}_{ij}^* + \mathbf{s}_{i0}^* \mathbf{x}_{ij}^* = 0 \quad \textit{for } i = 1, \ldots n \textit{ and } j = 1, \ldots n_i$$

where $\mathbf{x}_{ij}$ *and* $\mathbf{s}_{ij}$ *are respectively the* j*th entry of* $\mathbf{x}_i$ *and the* j*th entry of* $\mathbf{s}_i$*.*

2.2.1. Lagrange Multipliers for SOCP with Nonlinear Objective. Again, in applications we may encounter second-order cone programs with nonlinear but convex (or concave for maximization problem) objective functions. Let us state the Lagrangian theory for the case in which there is only one SOC inequality. Consider

$$\begin{array}{ll} \max & g(\mathbf{y}) \\ \text{s.t.} & \mathbf{c}^\top - \mathbf{y}^\top A \succcurlyeq_{\mathcal{Q}} \mathbf{0} \end{array} \tag{23}$$

with $g(\mathbf{y})$ a twice differentiable and concave function.

Now we can associate the Lagrange multiplier $\mathbf{x}$ to the SOC inequality and define the Lagrangian:

$$L(\mathbf{y}, \mathbf{x}) = g(\mathbf{y}) - \mathbf{x}^\top(\mathbf{c} - A^\top \mathbf{y}). \tag{24}$$

The first-order optimality condition for (23) can be stated as follows.

Theorem 2.6. *Assume that there exists* $\mathbf{y}$ *such that* $\mathbf{c}^\top - \mathbf{y}^\top A \succ_{\mathcal{Q}} \mathbf{0}$. *Then, a necessary condition for a feasible vector* $\mathbf{y}^*$ *to be an optimal solution of* (23) *is that there is a vector* $\mathbf{x}$ *such that the following relations hold:*

$$\nabla_{\mathbf{y}} L = \nabla_{\mathbf{y}} g(\mathbf{y}) - \mathbf{x}^\top A^\top = 0 \tag{25}$$

$$x_0(\mathbf{c} - A\mathbf{x})_i + x_i(\mathbf{c} - A\mathbf{x})_0 = 0 \tag{26}$$

$$\mathbf{x} \succcurlyeq_{\mathcal{Q}} 0 \tag{27}$$

where $\nabla_{\mathbf{y}} g(\cdot)$ *is the gradient of* $g(\mathbf{y})$.

2.3. Duality and Complementarity in General

The duality and complementarity results stated for SDP and SOCP actually extend to all convex optimization problems. Let $\mathcal{K}$ be a *proper cone*, namely

(1) $\mathcal{K}$ is a cone, that is, for all nonnegative $\alpha \geq 0$, if $\mathbf{x} \in \mathcal{K}$, then $\alpha\mathbf{x} \in \mathcal{K}$,
(2) $\mathcal{K}$ is closed (thus, it contains its boundary),
(3) $\mathcal{K}$ is convex, that is, for all $\mathbf{x}, \mathbf{y} \in \mathcal{K}$, $\mathbf{x} + \mathbf{y} \in \mathcal{K}$,
(4) $\mathcal{K}$ is pointed, that is, $\mathcal{K} \cap (-\mathcal{K}) = \{\mathbf{0}\}$, and
(5) $\mathcal{K}$ is full-dimensional, that is, relative interior of $\mathcal{K}$, in $\mathbb{R}^m$, written as $\text{Int}_m \mathcal{K}$, is nonempty.

Then any proper cone has a *dual cone* defined as

$$\mathcal{K}^* = \{\mathbf{y} \in \mathbb{R}^m \mid \mathbf{x}^\top \mathbf{y} \geq 0 \text{ for all } \mathbf{x} \in \mathcal{K}\}.$$

If $\mathcal{K}$ is a proper cone, then so is $\mathcal{K}^*$. Also note that $\left(\mathcal{K}^*\right)^* = \mathcal{K}$.

Now, consider the following pair of optimization problems.

$$\begin{array}{llll} \textit{Primal}: & & \textit{Dual}: & \\ \min & \mathbf{c}^\top \mathbf{x} & \max & \mathbf{b}^\top \mathbf{y} \\ \text{s.t.} & A\mathbf{x} = \mathbf{b} & \text{s.t.} & A^\top \mathbf{y} + \mathbf{s} = \mathbf{c} \\ & \mathbf{x} \in \mathcal{K} & & \mathbf{s} \in \mathcal{K}^* \end{array} \tag{28}$$

This pair of optimization problems are generalizations of linear, semidefinite, and second-order cone programming problems. In these special cases, the underlying cones $\mathcal{K}$ are the nonnegative orthant, the positive semidefinite matrices, and second-order cones, respectively. Also, in these three special cases, the underlying cones are self-dual, that is, for each of nonnegative orthant, semidefinite matrices, and second-order cones we have $\mathcal{K} = \mathcal{K}^*$. However,

in general, it is not the case that all cones are self-dual. Indeed, we will see an example of such cones below when we discuss positive polynomials. It is fairly straightforward to show that all convex optimization problems can be transformed into (28) with addition of extra variable and constraints.

As in the case of SDP and SOCP, weak duality is almost immediate:

$$\mathbf{c}^\top \mathbf{x} - \mathbf{b}^\top \mathbf{y} = \mathbf{c}^\top \mathbf{x} - (A\mathbf{x})^\top \mathbf{y} = \left(\mathbf{c}^\top - \mathbf{y}^\top A\right)\mathbf{x} = \mathbf{s}^\top \mathbf{x} \geq 0$$

where the last inequality is because $\mathbf{x} \in \mathcal{K}$ and $\mathbf{s} \in \mathcal{K}^*$. The strong duality also holds under certain sufficient conditions as stated in the following

Theorem 2.7. Strong Duality for Cone LP. *Let $\mathbf{x}, \mathbf{s} \in \mathbb{R}^m$, and let $\mathbf{y} \in \mathbb{R}^k$. Assume at least one of the following statements is true:*

- *There is a primal feasible vector $\mathbf{x} \in \operatorname{Int}_m \mathcal{K}$*
- *There are dual feasible vectors $\mathbf{y}$ and $\mathbf{s}$ with $\mathbf{s} \in \operatorname{Int}_m \mathcal{K}^*$.*

Then,

i. *If the primal problem is unbounded, that is, there is a sequence of feasible vectors $\mathbf{x}^{(k)}$ such that the value of the objective function $z_k = \mathbf{c}^\top \mathbf{x}^{(k)}$ tends to infinity as $k \to \infty$, then the dual problem is infeasible.*

ii. *If the dual problem is unbounded, that is, there is a sequence of feasible vectors $\mathbf{y}^{(k)}$ and vectors $\mathbf{s}^{(k)}$ such that the objective function $u_k = \mathbf{b}^\top \mathbf{y}^{(k)}$ tends to infinity as $k \to \infty$, then the primal problem is infeasible.*

iii. *If both primal and dual problems are feasible and bounded, then for the optimal primal solution $\mathbf{x}^*$ and the optimal dual solution $\mathbf{y}^*$ and $\mathbf{s}^*$, we have*

$$\mathbf{c}^\top \mathbf{x}^* = \mathbf{b}^\top \mathbf{y}^* \quad \textit{equivalently} \quad (\mathbf{x}^*)^\top \mathbf{s}^* = 0.$$

Once again, strong duality leads to complementary slackness theorem. However, in the general case, a nice set of equations as in SDP or SOCP may not be readily available. We can make the following statement though:

Lemma 2.3. *Let $\mathcal{K}$ and its dual $\mathcal{K}^*$ be proper cones in $\mathbb{R}^m$. Define the complementary set of $\mathcal{K}$ as*

$$C(\mathcal{K}) = \{(\mathbf{x}, \mathbf{y}) \mid \mathbf{x} \in \mathcal{K}, \mathbf{y} \in \mathcal{K}^*, \text{ and } \mathbf{x}^\top \mathbf{y} = 0\}.$$

Then $C(\mathcal{K})$ is an m-dimensional manifold homeomorphic to $\mathbb{R}^m$.

This lemma says that there are some m equations $f_i(\mathbf{x}, \mathbf{s}) = 0$ that characterize the set $C(\mathcal{K})$. For instance, if $\mathcal{K}$ is the cone of positive semidefinite matrices, then we saw that $\mathcal{C}(\mathcal{K})$ is characterized by the $m = n(n+1)/2$ equations $XY + YX = 0$. And in the case of second-order cone $\mathcal{Q}$, $m = n + 1$ and $C(\mathcal{Q})$ is characterized by the equations $\mathbf{x}^\top \mathbf{y} = 0$ and $x_0 y_i + y_0 x_i = 0$, for $i = 1, \ldots, n$. In general, for each cone we need to work out the *complementarity* equations $f_i(\mathbf{x}, \mathbf{y}) = 0$ individually. Finally, note that putting together primal and dual feasibility equations and the complementarity conditions we get the system of equations

$$\begin{aligned} &\mathbf{b} - A\mathbf{x} = \mathbf{0} \\ &\mathbf{c} - A^\top \mathbf{y} - \mathbf{s} = \mathbf{0} \\ &f_i(\mathbf{x}, \mathbf{s}) = 0, \quad \text{for } i = 1, \ldots, m. \end{aligned} \tag{29}$$

Due to the complementarity relations, this system of equations is now square; that is, the number of variables and equations are equal. Of course, many conditions need to be

satisfied for this system to be solvable. Writing this system succinctly as $\mathbf{F}(\mathbf{x},\mathbf{y},\mathbf{s})=\mathbf{0}$, there are classes of algorithms that generate a sequence of estimates $\left(\mathbf{x}^{(k)},\mathbf{y}^{(k)},\mathbf{s}^{(k)}\right)$ such that $\mathbf{F}\left(\mathbf{x}^{(k)},\mathbf{y}^{(k)},\mathbf{s}^{(k)}\right)$ tends to zero as $k\to\infty$.

3. Nonnegativity and Semidefinite Programming

In this section, we take up the study of nonnegative polynomials in one variable, and the more general multivariate polynomials that can be expressed as sum of squares of other polynomials. This area, as will be seen in the following sections, is important in approximation and regression of functions that in one way or another are bounded by other functions.

3.1. Nonnegative Polynomials and the Moment Cone

Polynomials and polynomial splines (to be defined in §4) are important in approximation and regression of unknown functions. In some cases, we may wish to approximate a nonnegative function, and it may be required that the approximating polynomial or polynomial spline also be nonnegative. Here, we study the cone linear programming problem over the cone of positive polynomials. Let us now formally define this cone and its dual. The cone of positive polynomials is

$$\mathcal{P}=\{\mathbf{p}=(p_0,p_1,\ldots,p_{2n})\mid p(t)=p_0+p_1t+\cdots+p_{2n}t^{2n}\geq 0 \text{ for all } t\in\mathbb{R}\}.$$

Also consider the so-called *moment cone* defined as follows

$$\mathcal{M}=\Big\{\mathbf{c}=(c_0,c_1,\ldots,c_{2n})\mid \text{there is } \alpha\geq 0, \text{ and a probability distribution function } F,\\ \text{where } c_i=\alpha\textstyle\int_{\mathbb{R}}t^i dF,\ i=0,\cdots,2n\Big\}\cup\{(0,0,\ldots,0,\beta)\mid\beta\geq 0\}.$$

$\mathcal{M}$ is the cone generated by all vectors that are moments of some probability distribution function. However, the moments alone are not enough to generate a closed cone. For instance, for any $\epsilon>0$, the vector $(1,\epsilon,1/\epsilon)$ is the moment vector of normal distribution with mean ϵ and variance ϵ^2-1/ϵ^2. Thus, for all ϵ, the vector $\mathbf{c}(\epsilon)=\epsilon(1,\epsilon,1/\epsilon)=(\epsilon,\epsilon^2,1)$ is in the moment cone. However, as $\epsilon\to 0$ the vector $\mathbf{c}(\epsilon)$ converges to $(0,0,1)$, which is not a nonnegative multiple of any vector of moments. This is why we include the ray $\alpha\mathbf{e}_n$ (where $\mathbf{e}_n=(0,0,\ldots,0,1)$) and with that $\mathcal{M}$ becomes a closed cone.

Define $\mathbf{u}_t=(1,t,t^2,\ldots,t^{2n})$. It can be shown that for every $\mathbf{c}\in\mathcal{P}$ there are at most n distinct real numbers $t_1,\ldots,t_n$ and n nonnegative real numbers $\alpha_1,\ldots,\alpha_n$ such that $\mathbf{c}=\sum_i\alpha_i\mathbf{u}_{t_i}$. In fact, the vectors $\mathbf{u}_t$ along with $\mathbf{e}_n$ make up all the extreme rays of $\mathcal{M}$. For each $\mathbf{u}_t$ of length $2n+1$, define the matrix

$$U_t=\begin{pmatrix}1\\ t\\ t^2\\ \vdots\\ t^n\end{pmatrix}(1,t,t^2,\ldots,t^n)=\begin{pmatrix}1 & t & t^2 & \cdots & t^n\\ t & t^2 & t^3 & \cdots & t^{2n+1}\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ t^n & t^{n+1} & t^{n+2} & \cdots & t^{2n}\end{pmatrix}.$$

This rank-one matrix is in fact a *Hankel* matrix; that is, it is constant along its *reverse diagonals*. Because any linear combination of Hankel matrices is again a Hankel matrix, it follows that any moment vector is uniquely represented by a positive semidefinite Hankel matrix. In fact, we have

Theorem 3.1. *The vector* $\mathbf{c} = (c_0, c_1, \ldots, \mathbf{c}_{2n}) \in \mathcal{M}$ *if and only if the Hankel matrix*

$$H(\mathbf{c}) = \begin{pmatrix} c_0 & c_1 & c_2 & \cdots & c_n \\ c_1 & c_2 & c_3 & \cdots & c_{2n+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_n & c_{n+1} & c_{n+2} & \cdots & c_{2n} \end{pmatrix}$$

is positive semidefinite.

Now, let us examine the dual cone $\mathcal{M}^*$, which by definition consists of all vectors $p_0, p_1, \ldots, p_{2n}$ such that $\mathbf{p}^\top \mathbf{c} \geq 0$ for all $\mathbf{c} \in \mathcal{M}$. In particular, for every t,

$$\mathbf{p}^\top \mathbf{u}_t = p_0 + p_1 t + \cdots + p_{2n} t^{2n} \geq 0.$$

Thus, all nonnegative polynomials are included in $\mathcal{M}^*$. It is a simple matter to show that $\mathcal{M}^* = \mathcal{P}$.

From the matrix representation of moment vectors, one can find a matrix representation for positive polynomials:

Theorem 3.2. *A polynomial* $p(t)$ *represented by its vector of coefficients* $\mathbf{p} = (p_0, p_1, \ldots, p_{2n})$ *is nonnegative for all* t *if and only if there is a positive semidefinite matrix*

$$Y = \begin{pmatrix} Y_{00} & Y_{01} & \cdots & Y_{0n} \\ Y_{10} & Y_{11} & \cdots & Y_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n0} & Y_{n1} & \cdots & Y_{nn} \end{pmatrix}$$

such that

$$p_k = Y_{0k} + Y_{1,k-1} + \cdots + Y_{k0} \quad \textit{for } k \leq n$$

$$p_k = Y_{kn} + Y_{k+1,n-1} + \cdots + Y_{n,k} \quad \textit{for } k > n$$

for $k = 0, 1, \ldots, 2n$.

Some observations about nonnegative polynomials are in order. If a nonnegative polynomial has a root, then that root must have an even multiplicity; otherwise, in some neighborhood of that root, it will dip below zero. If a polynomial is strictly positive, then all of its roots are complex numbers, and because the polynomial has real coefficients, the complex roots appear in conjugate pairs. Noting that $(t - a - bi)(t - a + bi) = (t - a)^2 + b^2$, we conclude that a polynomial $p(t)$ of degree $2n$ is nonnegative if and only if

$$p(t) = \alpha (t - t_1)^2 \cdots (t - t_k)^2 \prod_{i=1}^{s} ((t - \alpha_i)^2 + \beta_i)$$

where either of k or s can be zero, and $\alpha > 0$ is the coefficient of the highest power term of the polynomial. From this observation, it is not difficult to show that a polynomial is nonnegative if and only if it is a nonnegative sum of polynomials that are square and with only real roots.

Theorem 3.3. *The extreme rays of* $\mathcal{P}$ *are coefficients of polynomials of the form*

$$p_{2r} \prod_{i=1}^{r} (t - t_i)^2, \quad p_{2r} > 0.$$

When $\mathbf{c} \in \mathcal{M}$ and $\mathbf{p} \in \mathcal{P}$ and $\mathbf{p}^\top \mathbf{c} = 0$ then, as we saw earlier for arbitrary proper cones, there must exist $2n+1$ equations relating $\mathbf{p}$ and $\mathbf{c}$. We may derive some of these equations relatively easily: If $\mathbf{c} = \sum_{k=1}^{r} \alpha_k \mathbf{u}_{t_k}$, then

$$0 = \mathbf{p}^\top \mathbf{c} = \sum_{k=1}^{r} \mathbf{p}^\top \mathbf{u}_{t_k} = \sum_{k} p(t_k).$$

Because each $p(t_k) \geq 0$ and they add up to 0, then each of them must be 0; that is, $p(t_i) = 0$. On the other hand, each $\mathbf{p}$ can be written as $\sum_{i=1}^{s} \beta_i \mathbf{p}_i$, where the polynomials $p_i(t)$ have only real roots of even multiplicity. Thus, $\mathbf{p}^\top \mathbf{c} = 0$ implies $p_j(t_i) = 0$ for $j = 1, \ldots, s$ and $i = 1, \ldots, r$.

3.1.1. Nonnegative Polynomials and Moments Over an Interval. In most applications we are actually interested in polynomials that are nonnegative over an interval $[a, b]$. It is still true that $\mathcal{P}([a,b])$, the cone of polynomials nonnegative on $[a,b]$, is the dual of $\mathcal{M}([a,b])$, the cone of moments where the distribution is concentrated on $[a,b]$. More precisely

$$\mathcal{M}([a,b]) = \big\{\mathbf{c} = (c_0, c_1, \ldots, c_{2n}) \,|\, \text{there is } \alpha \geq 0, \text{ and a probability distribution function } F, \\ \text{where } c_i = \alpha \textstyle\int_a^b t^i dF, i = 0, \cdots, 2n\big\}.$$

Note that in this case, the cone generated by moments need not be augmented because it is already closed.

The matrix characterization of $\mathcal{M}([a,b])$ and $\mathcal{P}([a,b])$ are similar to the case which the interval was all of $\mathbb{R}$, except that it is a bit more complex. As before, we represent a polynomial $p_0 + p_1 x + \cdots + p_n x^n$ by its vector of coefficients $\mathbf{p} = (p_0, p_1, \ldots, p_n)$. Then, $\mathcal{M}([a,b])$ can be expressed by defining the following matrices:

$$\underline{H}_{2m} = (c_{i+j})_{ij}, \qquad 0 \leq i, j \leq m \tag{30}$$

$$\overline{H}_{2m} = \big((a+b)c_{i+j+1} - c_{i+j+2} - abc_{i+j}\big)_{ij}, \qquad 0 \leq i, j \leq m-1 \tag{31}$$

$$\underline{H}_{2m+1} = \big(c_{i+j+1} - ac_{i+j}\big)_{ij}, \qquad 0 \leq i, j \leq m \tag{32}$$

$$\overline{H}_{2m+1} = \big(bc_{i+j} - c_{i+j+1}\big)_{ij}, \qquad 0 \leq i, j \leq m. \tag{33}$$

From the analysis in Dette and Studden [7], Karlin and Studden [10], and Nesterov [12] the matrices defined by (30)–(33) are related to the moment cone as follows:

$$\text{when } n = 2m, \quad (c_0, c_1, \ldots, c_n) \in \mathcal{M}([a,b]) \quad \text{iff} \quad \underline{H}_{2m} \succcurlyeq 0 \text{ and } \overline{H}_{2m} \succcurlyeq 0, \tag{34}$$

$$\text{when } n = 2m+1, \quad (c_0, c_1, \ldots, c_n) \in \mathcal{M}([a,b]) \quad \text{iff} \quad \underline{H}_{2m+1} \succcurlyeq 0 \text{ and } \overline{H}_{2m+1} \succcurlyeq 0. \tag{35}$$

Let E_k^m, be the $(m+1) \times (m+1)$ matrix given by

$$(E_k^m)_{ij} = \begin{cases} 1, & i+j = k \\ 0, & i+j \neq k, \end{cases} \qquad 0 \leq i, j \leq m.$$

Then $E_0^m, E_1^m, \ldots, E_{2m}^m$ form a basis for the space of $(m+1) \times (m+1)$ Hankel matrices. We may omit the superscript m where it is fixed from context, and write simply E_k.

Using (34) and (35), we can now characterize the cone $\mathcal{M}([a,b])$ and its dual $\mathcal{P}([a,b])$. The details differ depending on whether n is odd or even, and thus whether we employ (34) or (35).

When $n = 2m+1$: Rewriting (32) and (33) in terms of the basis elements $E_0, \ldots, E_{2m+1}$, we have

$$\underline{H}_{2m+1} = -c_0 a E_0 + c_1(E_0 - aE_1) + c_2(E_1 - aE_2) + \cdots + c_{2m}(E_{2m-1} - aE_{2m}) + c_{2m+1}E_{2m}$$
$$\overline{H}_{2m+1} = c_0 b E_0 + c_1(bE_1 - E_0) + \cdots + c_{2m}(bE_{2m} - E_{2m-1}) - c_{2m+1}E_{2m}.$$

Therefore, re-expressing the positive semidefiniteness conditions in (34), the cone $\mathcal{M}_{n+1}$ consists of all vectors $(c_0, c_1, \ldots, c_n)$, satisfying

$$-c_0 a E_0 + c_1(E_0 - aE_1) + \cdots + c_{2m}(E_{2m-1} - aE_{2m}) + c_{2m+1}E_{2m} \succcurlyeq 0 \tag{36}$$
$$c_0 b E_0 + c_1(bE_1 - E_0) + \cdots + c_{2m}(bE_{2m} - E_{2m-1}) - c_{2m+1}E_{2m} \succcurlyeq 0. \tag{37}$$

To characterize dual cone $\mathcal{P}([a,b])$, we associate symmetric positive semidefinite matrices X and Y with (36) and (37), respectively. These matrices play much the same role as Lagrange multipliers in general nonlinear programming, except that they must be matrices of the same shape as the two sides of the semidefinite inequalities (36)–(37), that is, both X and Y are $(m+1) \times (m+1)$ symmetric matrices. Using the inner product of matrices defined in §1, we then argue that $(p_0, p_1, \ldots, p_n)$ is in $\mathcal{P}([a,b])$ whenever

$$\begin{aligned}
p_0 &= -aE_0 \bullet X + bE_0 \bullet Y \\
p_1 &= (E_0 - aE_1) \bullet X + (bE_1 - E_0) \bullet Y \\
p_2 &= (E_1 - aE_2) \bullet X + (bE_2 - E_1) \bullet Y \\
&\vdots \\
p_k &= (E_{k-1} - aE_k) \bullet X + (bE_k - E_{k-1}) \bullet Y \\
&\vdots \\
p_{2m+1} &= E_{2m} \bullet X - E_{2m+1} \bullet Y.
\end{aligned} \tag{38}$$

When $n = 2m$: In the case where n is even, we can apply a similar analysis to (35), resulting in the characterization that $(p_0, \ldots, p_n) \in \mathcal{P}_{n+1}(a,b)$ if and only if

$$\begin{aligned}
p_0 &= E_0^m \bullet X - abE_0^{m-1} \bullet Y \\
p_1 &= E_1^m \bullet X + ((a+b)E_0^{m-1} - abE_1^{m-1}) \bullet Y \\
p_2 &= E_2^m \bullet X + (-E_0^{m-1} + (a+b)E_1^{m-1} - abE_2^{m-1}) \bullet Y \\
&\vdots \\
p_k &= E_k^m \bullet X + (-E_{k-2}^{m-1} + (a+b)E_{k-1}^{m-1} - abE_k^{m-1}) \bullet Y \\
&\vdots \\
p_{2m} &= E_{2m}^m \bullet X - E_{2m-2}^{m-1} \bullet Y \\
X &\succcurlyeq 0 \\
Y &\succcurlyeq 0,
\end{aligned}$$

where the symmetric matrices X and Y have dimension $(m+1) \times (m+1)$ and $m \times m$, respectively.

3.1.2. Cubic Polynomials with Shifted Representations. The special case of cubic polynomials is of particular interest, because they are the most common form of splines used in practice. In this section, we present the details of matrix representations of nonnegative cubic polynomials over an interval $[a,b]$.

Sometimes it is convenient to represent a nonnegative polynomial over $[a,b]$ by $p(x) = p_0 + p_1(x-a) + p_2(x-a)^2 + \cdots + p_n(x-a)^n$. In this case, because $p(x)$ is nonnegative over $[a,b]$ if and only if $p_0 + p_1 t + p_2 t^2 + \cdots + p_n t^n$ is nonnegative over $[0, b-a]$, the representations given above can be modified by replacing a with 0 and b with $d = b-a$.

In particular, consider the cone $\mathcal{P}([0,d])$ of cubic polynomials $p(t) = p_0 + p_1(t-a) + p_2(t-a)^2 + p_3(t-a)^3$ that are nonnegative over $[a,b]$. First, specializing (36) and (37) to $m=1$, and replacing $a \leftarrow 0$ and $b \leftarrow d$, we note that a vector (c_0, c_1, c_2, c_3) is in the dual cone $\mathcal{M}([0,d])$ if and only if

$$\begin{pmatrix} c_1 & c_2 \\ c_2 & c_3 \end{pmatrix} \succcurlyeq 0 \quad \text{and} \quad \begin{pmatrix} dc_0 - c_1 & dc_1 - c_2 \\ dc_1 - c_2 & dc_2 - c_3 \end{pmatrix} \succcurlyeq 0.$$

Specializing the Lagrange multiplier analysis for the $n = 2m+1$ case above, the cubic polynomial $p_0 + p_1(t-a) + p_2(t-a)^2 + p_3(t-a)^3$ is nonnegative on $[a,b]$ whenever there are 2×2 matrices

$$X = \begin{pmatrix} x & y \\ y & z \end{pmatrix} \quad \text{and} \quad Y = \begin{pmatrix} s & v \\ v & w \end{pmatrix}$$

satisfying

$$\begin{aligned}
p_0 = dE_0 \bullet Y \quad &\Longleftrightarrow \quad p_0 = ds \\
p_1 = E_0 \bullet X + (dE_1 - E_0) \bullet Y \quad &\Longleftrightarrow \quad p_1 = x + 2dv - s \\
p_2 = E_1 \bullet X + (dE_2 - E_1) \bullet Y \quad &\Longleftrightarrow \quad p_2 = 2y + dw - 2v \\
p_3 = E_2 \bullet X + -E_2 \bullet Y \quad &\Longleftrightarrow \quad p_3 = z - w \\
X \succcurlyeq 0 \quad &\Longleftrightarrow \quad x, z \geq 0, \quad \mathrm{Det}(X) = xz - y^2 \geq 0 \\
Y \succcurlyeq 0 \quad &\Longleftrightarrow \quad s, w \geq 0, \quad \mathrm{Det}(Y) = sw - v^2 \geq 0.
\end{aligned}$$

In this case, because of the low dimension of X and Y, the positive semidefiniteness constraints $X, Y \succcurlyeq 0$ can be reformulated as the simple linear and quadratic constraints $x, z, s, w \geq 0$, $xz - y^2 \geq 0$, and $sw - v^2 \geq 0$, all of which are in fact SOC inequalities. Thus, the nonnegativity constraints for cubic polynomials can be expressed by two SOC constraints and four simple nonnegativity constraints.

3.2. Other Moments and Polynomials

Here, we briefly mention that trigonometric polynomials and moments are also semidefinite representable. Briefly, a trigonometric polynomial of degree n is a linear combination of functions in

$$\{1, \cos(t), \sin(t), \cos(2t), \sin(2t), \ldots, \cos(nt), \sin(nt)\}.$$

Then, the cone of nonnegative trigonometric polynomials is a proper cone in $\mathbb{R}^{2n+1}$. As in the case of ordinary polynomials, the dual cone is given by

$$\begin{aligned}
\mathcal{M} = \mathrm{cl}\big\{ \mathbf{c} = (c_0, c_1, \ldots, c_{2n}) \mid &\text{there is } \alpha \geq 0, \text{ and a probability distribution function } F, \\
&\text{where } c_i = \alpha \textstyle\int_{\mathbb{R}} \cos(it)\, dF, \text{ if } i \text{ is odd, and } c_i = \alpha \int_{\mathbb{R}} \sin(it)\, dF \text{ if } i \text{ is even} \big\}.
\end{aligned}$$

It turns out that instead of Hankel matrices, the trigonometric polynomials use positive semidefinite *Töplitz* matrices. A characterization analogous to ordinary polynomials exists for nonnegative trigonometric polynomials. Similar characterization also holds for trigonometric polynomials over interval $[a, b]$.

Finally, the concept of positive polynomials can be generalized. A set of functions $\{f_1(t), f_2(t), \ldots, f_n(t)\}$ satisfying

- $f_i(t)$ are linearly independent, and
- any equation of the form $\sum_{i=1}^{n} p_i f_i(t) = 0$ has at most $n+1$ zeros (except the identically zero function, of course),

is called a *Chebyshev system*. Within the Chebyshev system, one can speak of *polynomials* to mean any function $p(t) = \sum_i p_i f_i(t)$. And within this linear space of functions, one can consider the cone of nonnegative polynomials, and the dual cone of moments (which is generated by the vectors of means of $f_i(t)$ with respect to one common distribution function).

It is not known whether all these cones are semidefinite representable. However, Faybusovich [8] has developed a straightforward optimization method over such cones, by showing how to compute a *barrier function* for them (see §5 below).

3.3. Cones Generated by Sum of Squares of Functions

A generalization of the class of positive univariate polynomials is the set of functions that can be expressed as sum of squares of a given class of functions. It was shown by Nesterov [12] that this class of functions is also semidefinite representable.

Let $S = \{u_1(\mathbf{x}), \ldots, u_n(\mathbf{x})\}$ be a set of linearly independent functions over some domain $\Delta \subset \mathbb{R}^k$. We wish to characterize the cone

$$\mathcal{T} = \left\{ \sum_{i=1}^{N} p_i^2(\mathbf{x}) \;\middle|\; p_i(\mathbf{x}) \in \operatorname{span} S \right\} \tag{39}$$

where $N \geq n$ is a fixed number. This cone is convex. We now discuss Nesterov's construction to show that $\mathcal{T}$ is semidefinite representable. Define

$$S^2 = \{u_i(\mathbf{x})u_j(\mathbf{x}) \mid 1 \leq i, j \leq n\}.$$

Also, let $\mathbf{v}(\mathbf{x}) = (v_1(\mathbf{x}), \ldots, v_m(\mathbf{x}))^\top$ be a vector whose entries form a basis of $\mathcal{L}_m = \operatorname{span} S^2$. Then, for each of elements $u_i(\mathbf{x})u_j(\mathbf{x})$ in S^2 there is a vector $\boldsymbol{\lambda}_{ij} \in \mathcal{L}_m$ such that

$$u_i(\mathbf{x})u_j(\mathbf{x}) = \boldsymbol{\lambda}_{ij}^\top \mathbf{v}(\mathbf{x}).$$

The $\boldsymbol{\lambda}_{ij}$'s together define a linear mapping, sending $\mathbf{c} \in \mathcal{L}_m$ to the symmetric matrix $\Lambda(\mathbf{c})$ with ij entry equal to $\boldsymbol{\lambda}_{ij}^\top \mathbf{x}$. Let us assume that $\Lambda(\mathbf{c}) = \sum_i c_i F_i$; that is, F_i's, are a basis of the linear space $\Lambda(\mathcal{L}_m)$. Note that in particular $\Lambda(\mathbf{v}(\mathbf{x})) = \mathbf{v}(\mathbf{x})\mathbf{v}(\mathbf{x})^\top$, a symmetric rank-one positive semidefinite matrix. Then, the main result about the semidefinite representation of $\mathcal{T}$ is the following.

Theorem 3.4. (Nesterov [12]).
(1) *The cone $\mathcal{T}^*$, the dual cone of sum-of-squares functional system, is a proper cone characterized by*

$$\mathcal{T}^* = \{\mathbf{c} \in \mathbb{R}^m \mid \Lambda(\mathbf{c}) \succcurlyeq 0\}.$$

(2) $\mathcal{T}$ *is also a proper cone characterized as follows: Let* $\mathbf{p}(\mathbf{x}) \in \mathcal{T}$ *be represented by its vector of coefficients* $\mathbf{p} \in \mathbb{R}^m$. *Then,*

$$\mathcal{T} = \{\mathbf{p} \mid \textit{there is a symmetric } n \times n \textit{ matrix } Y \succcurlyeq 0,\ Y \bullet F_i = p_i,\ i = 1, \ldots, n\}.$$

Example 3.1. Sum of Squares of Biquadratic Functions of Two Variables. Let $\mathbf{x} = (t, s)$ and $S = \{u_1 = 1, u_2 = t, u_3 = t^2, u_4 = s, u_5 = s^2, u_6 = ts\}$; thus span S is the set of all linear, quadratic, and bilinear functions in variables s and t. Then

$$S^2 = \{1, t, t^2, s, s^2, ts, t^3, ts^2, t^2 s, t^4, t^2 s^2, t^3 s, s^3, s^4, ts^3\}$$

with duplicates removed. Taking S^2 as the basis, we see that $\mathcal{T}^*$ is a 15-dimensional cone made up of vectors $\mathbf{c} = (c_1, \ldots, c_{15})$ such that

$$\begin{pmatrix} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \\ c_2 & c_3 & c_7 & c_6 & c_8 & c_9 \\ c_3 & c_7 & c_{10} & c_9 & c_{11} & c_{12} \\ c_4 & c_6 & c_9 & c_5 & c_{13} & c_8 \\ c_5 & c_6 & c_{11} & c_{13} & c_{14} & c_{15} \\ c_6 & c_9 & c_{12} & c_8 & c_{15} & c_{11} \end{pmatrix} \succcurlyeq 0.$$

Now the set of polynomials of variables t and s that are sum of squares of polynomials is span S are represented by the coefficients $\mathbf{p} = (p_1, p_2, \ldots, p_{15})^\top$ where

$$\begin{aligned} p_1 + p_2 t + p_3 t^2 + p_4 s + p_5 s^2 + p_6 ts + p_7 t^3 + p_8 ts^2 + p_9 t^2 s + p_{10} t^4 \\ + p_{11} t^2 s^2 + p_{12} t^3 s + p_{13} s^3 + p_{14} s^4 + p_{15} ts^3 \geq 0 \quad \text{for all } t, s. \end{aligned}$$

Then, $\mathcal{T}$ consists of those vectors $\mathbf{p} = (p_1, \ldots, p_{15})$ such that there is a 6×6 positive semidefinite matrix Y where

$$\begin{aligned} p_1 &= Y_{1,1}, p_2 = Y_{1,2}, p_3 = Y_{1,3} + Y_{2,2}, p_4 = Y_{1,4}, p_5 = Y_{1,5} + Y_{4,4}, \\ p_6 &= Y_{1,6} + Y_{2,4}, p_7 = Y_{2,3}, p_8 = Y_{2,5} + Y_{4,6}, p_9 = Y_{2,6} + Y_{3,4}, \\ p_{10} &= Y_{3,3}, p_{11} = Y_{3,5} + Y_{6,6}, p_{12} = Y_{3,6}, p_{13} = Y_{4,5}, \\ p_{14} &= Y_{5,5}, p_{15} = Y_{5,6}. \end{aligned}$$

It is possible to generalize this characterization to a *weighted* sum of squares, provided that the weights $q_i(\mathbf{x})$ are given fixed functions. Let the functions $q_1(\mathbf{x}), \ldots, q_l(\mathbf{x})$ be all nonnegative on $\Delta \subseteq \mathbb{R}^k$. And let $S_1, \ldots, S_l$ be l sets containing function $u_{ij}(\mathbf{x})$ where $i = 1, \ldots, l$ and $j = 1, \ldots, n_i$. Now define

$$\mathcal{T}(q_1, \ldots, q_l) = \left\{ \sum_{j=1}^{l} q_j(\mathbf{x}) \sum_{i=1}^{N} \mathbf{p}_{ij}^2(\mathbf{x}) \;\middle|\; p_{ij}(\mathbf{x}) \in S_i \right\}. \tag{40}$$

Then, $\mathcal{T}^*(q_1, \ldots, q_l)$ consists of vectors $\mathbf{c} \in \mathbb{R}^m$ such that $\Lambda_i(\mathbf{c}) \succcurlyeq 0$. Here each Λ_i is defined relative to S_i the same way Λ was defined relative to S above. Because each $\Lambda_i(\mathbf{c})$ is a matrix-valued operator linearly dependent on $\mathbf{c}$, there are matrices F_{ij} such that $\Lambda_i = \sum_j c_j F_{ij}$. Then, the cone $\mathcal{T}(q_1, \ldots, q_l)$ can be expressed as

$$\mathbf{p} \in \mathcal{T}(q_1, \ldots, q_l) \iff \text{there are } Y_i \succcurlyeq 0 \text{ such that } \sum_i F_{ij} \bullet Y_i = p_j.$$

Example 3.2. Weighted Sum of Biquadratics Over a Triangle. Let Δ be the triangle in $\mathbb{R}^2$ with sides $x \geq 0, 1-y \geq 0$, and $x-y \geq 0$; that is, $q_1(x,y) = (x-y)$, $q_2(x,y) = x$, and $q_3(x,y) = y$. Define

$$\begin{aligned} S_1 &= \{1, x, y\}, & \mathbf{v}_1(x,y) &= (1, x, y, x^2, xy, y^2) \\ S_2 &= \{1, x, y, y^2\} & \mathbf{v}_2(x,y) &= (1, x, y, y^2, x^2, xy, xy^2, y^3, y^4) \quad \text{and} \\ S_3 &= \{1, x, x^2, y\} & \mathbf{v}_3(x,y) &= (1, x, x^2, y, x^3, xy, x^4, x^2y, y^2). \end{aligned}$$

Then, similar calculations to Example 3.3 yields

$$\Lambda_1(\mathbf{c}) = \begin{pmatrix} c_1 & c_2 & c_3 \\ c_2 & c_4 & c_5 \\ c_3 & c_5 & c_6 \end{pmatrix} \quad \Lambda_2(\mathbf{c}) = \begin{pmatrix} c_1 & c_2 & c_3 & c_6 \\ c_2 & c_6 & c_2 & c_7 \\ c_3 & c_2 & c_6 & c_8 \\ c_6 & c_7 & c_8 & c_9 \end{pmatrix} \quad \text{and} \quad \Lambda_3(\mathbf{c}) = \begin{pmatrix} c_1 & c_2 & c_3 & c_4 \\ c_2 & c_4 & c_{10} & c_5 \\ c_3 & c_{10} & c_{11} & c_{12} \\ c_4 & c_5 & c_{12} & c_6 \end{pmatrix}.$$

Now, a polynomial $p_1 + p_2x + p_3y + p_4x^2 + p_5xy + p_6y^2 + p_7xy^2 + p_8y^3 + p_9y^4 + p_{10}x^3 + p_{11}x^4 + p_{12}x^2y$ is a weighted sum of squares with weights $(x-y), x, y$ over the triangle if there is a 3×3 matrix X and two 4×4 matrices Y and Z such that

$$\begin{aligned} p_1 &= X_{1,1} + Y_{1,1} + Z_{1,1}, & p_2 &= X_{1,2} + X_{2,1} + Y_{1,2} + Y_{2,1} + Z_{1,2} + Z_{2,1}, \\ p_3 &= X_{1,3}X_{3,1} + Y_{1,3} + Y_{3,1} + Z_{1,3} + Z_{3,1}, & p_4 &= x_{2,2} + z_{2,2}, \\ p_5 &= X_{2,3} + X_{3,2} + Z_{2,4} + Z_{4,2}, & p_6 &= X_{3,3} + Y_{3,3} + Z_{4,4}, \quad p_7 = Y_{2,4} + Y_{4,2}, \\ p_8 &= Y_{3,4} + Y_{4,3}, & p_9 &= Y_{4,4}, \\ p_{10} &= Z_{2,3} + Z_{3,2}, & p_{11} &= Z_{3,3}, \quad p_{12} = Z_{3,4} + Z_{4,3}. \end{aligned}$$

Such weighted sums may be useful for *thin plate spline approximations* over plane.

4. Applications in Regression and Density Estimation

In this section, we will discuss applications of SDP and SOCP to a class of approximation and regression problems. Assume that we have a set of data or observations that arise from an unknown function $f(\mathbf{x})$. We assume that the (possibly multivariate) function $f(\mathbf{x})$ is continuous and differentiable up to order k, where k is a fixed integer (possibly equal to zero). Our goal is to approximate $f(\mathbf{x})$ from data "closely" according to some criterion for closeness. In addition, we require that either $f(\mathbf{x})$ or some linear functional of it be nonnegative.

It is this last requirement that is the point of departure from elementary approximation and regression theory. Furthermore, the nonnegativity condition on f or a linear functional of it can potentially connect the problem to SDP by restricting the set of eligible functions to nonnegative polynomials. We are using the term "polynomial" as a linear combination of a set of linearly independent functions. Of course, SDP is not the only way to approach "shape-constrained" and sign-restricted approximation and regression problems. However, in this section, we present one common approach that, along with the requirement of nonnegativity, leads to SDP or in an important particular case to SOCP.

First, let us indicate some of the problems of interest. Recall that the *Sobolev-Hilbert* space $S_m(\Delta)$ is the set of all functions defined on the domain $\Delta \subseteq \mathbb{R}^k$ with the property that all functions $f(\mathbf{x}) \in S_m(\Delta)$ are absolutely continuous, and have absolutely continuous

derivatives[1] $D^{\mathbf{r}} f$ of all orders up to $m-1$.[2] Furthermore, the derivatives of order m are square integrable over Δ. This space is endowed with an inner product defined as follows:

$$\langle f, g \rangle = \int_{\Delta} \sum_{\mathbf{r}} (D^{\mathbf{r}} f)(D^{\mathbf{r}} g)\, d\mathbf{x} \tag{41}$$

where the sum is taken over all nonnegative integer valued vectors $\mathbf{r}$ where $\sum_i r_i \leq m$.

The space $S_m(\Delta)$ can be closely approximated by polynomial splines of order m to arbitrary precision. We refer the reader to the texts of Chui [5] and Wahba [23] for multivariate splines, and content ourselves here with polynomial splines over an interval $[a, b]$. A polynomial spline of order m with knot vector $\mathbf{t} = (t_1, \ldots, t_s)$, $a \leq t_1 < t_2 < \cdots < t_s \leq b$, is a function $f(t)$ with the following properties:

- $f(t)$ is a polynomial of degree at most m on each open interval (t_i, t_{i+1}), and
- $f(t)$ is continuous and all derivatives of order up to $m-1$ are continuous.

It is well known that splines of order m with arbitrary fine-grid knot sequences are dense in $S_m([a, b])$. On the other hand, spline functions possess convenient computational properties. As a result, they are favored tools of both numerical analysts and statisticians for estimating unknown functions from a finite sample of data.

Within $S_m(\Delta)$, let $\mathcal{P}\big(S_m(\Delta)\big)$ be the cone of nonnegative functions. Consider the following classes of problems.

4.1. Parametric Linear Shape-Constrained Regression

We are given a set of data $(y_1, \mathbf{x}_1), \ldots, (y_n, \mathbf{x}_n)$, and we assume they are drawn from a model described by

$$y_i = f(\mathbf{x}) = \sum_j \theta_j f_j(\mathbf{x}_i) + \epsilon_i$$

where ϵ_i are i.i.d. random errors. In addition, given a linear functional $\mathcal{A}$, we must have that the function $\mathcal{A}f(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \Delta$. Our goal is to estimate the parameters θ_j in such a way that the estimator function is also nonnegative. Let us assume that the criteria for closeness is the *least squares* measure. Thus, our objective is to minimize $\sum_i \big(y_i - \sum_j \theta_j f_j(\mathbf{x}_i)\big)^2$.

There are several variations on this problem. First, consider the single variable case, that is the problem of estimating $f(x)$ when x and all the sample points x_i are in $[a, b]$. In addition, we may require that $f(x)$ be nonnegative, nondecreasing, nonincreasing, convex, and concave. All conditions can be expressed by nonnegativity of the first or second derivative of $f(\cdot)$. It is also possible to require that it be unimodal, but the mode needs to be specified (or we may have to conduct a one-dimensional search for it). If the functions $f_j(t)$ are polynomials or trigonometric polynomials, then all of these problems reduce to optimization over the cone of nonnegative polynomials. This assertion is the result of the simple observation that for ordinary (respectively, trigonometric) polynomials derivatives are also ordinary (respectively, trigonometric) polynomials. Let us look at an example:

Example 4.1. Least Square Approximation of a Smooth Concave Function. Let $(y_1, x_1), \ldots, (y_n, x_n)$ be a set of data drawn from a smooth function $f(x)$ over an interval $[a, b]$ with $f(a) = f(b) = 0$ and $f(\cdot)$ a concave function on $[a, b]$. Furthermore, suppose that $f(\cdot)$ is a polynomial of fixed degree, say of degree m. If we represent $f(x)$ by its vector of coefficients $\mathbf{f}$, then $f(x) = f_0 + f_1 x + \cdots + f_m x^m$. In this case, the role of θ_j are played by f_j. First, notice that the nonlinear objective function $\sum_{i=1}^n (y_i - \mathbf{f}^\top \mathbf{u}_{x_1})^2$ can be easily

[1] By $D^{\mathbf{r}} f$ where $\mathbf{r} = (r_1, \ldots, r_k)$ and $\sum_i r_i = r$, we mean any partial derivative $\partial^r f / \partial x_1^{r_1} \cdots \partial x_k^{r_k}$. Each r_i here is a nonnegative integer.

[2] Here, we mean the distributional sense of the term "derivative." Otherwise, if we use the ordinary definition, then we must subsequently complete the space to get a Hilbert space.

modeled using SOCP. In fact, we can replace the objective with a single variable z and add the constraint $z^2 \geq \sum_{i=1}^{n}(y_i - \mathbf{f}^\top \mathbf{u}_{x_1})^2$, which is an SOC constraint. For $f(\cdot)$ to be concave, its second derivative has to be nonpositive. Thus, our problem can be formulated as

$$\begin{aligned} \min \quad & z \\ \text{s.t.} \quad & (z, y_1 - \mathbf{f}^\top \mathbf{u}_{x_1}, \ldots, y_n - \mathbf{f}^\top \mathbf{u}_{x_n}) \in \mathcal{Q} \\ & \mathbf{f}^\top \mathbf{u}_a = \mathbf{f}^\top \mathbf{u}_b = 0 \\ & -(2, 6f_3, \ldots, k(k-1)f_k, \ldots, m(m-1)f_m) \in \mathcal{P}([a,b]) \end{aligned} \tag{42}$$

where, as before, $\mathbf{u}_a = (1, a, a^2, \ldots, a^m)$, and $\mathcal{P}([a,b])$ is the cone of nonnegative polynomials over the the interval $[a,b]$. The condition that a vector is in $\mathcal{P}([a,b]) \subseteq \mathbb{R}^{m-2}$ can be described by a pair of semidefinite constraints as described in §3.1. We should mention that if the polynomial degree is even moderately large, say larger than eight, then problem (42) is quite ill conditioned from a numerical point of view. It is advisable, therefore, to choose, instead of $1, t, t^2, \ldots$ a different basis with more favorable numerical characteristics for linear space of polynomials. For instance, we could use a sequence of orthogonal polynomials such as Chebyshev, Bernstein, Hermite, Laguerre, Legendre, etc., as our basis. In this case, the polynomial $f(t)$ can be written as a weighted sum of squares and therefore can be expressed by a pair of semidefinite constraints. This new formulation will have much better numerical properties and can be used to solve polynomials with quite large degrees.

For the multivariate case, characterization of nonnegative polynomials is computationally intractable (in fact, it is an NP-hard to decide whether a multivariate polynomial is nonnegative or not). However, it still may be possible to use the results of §3 and calculate a sum-of-squares (or weighted-sum-of-squares) polynomial approximation of nonnegative functions, provided that the function $f_j(\mathbf{x})$ are in the span of S^2 for some set of linearly independent functions S. Other shape-constrained requirements in the multivariate case can be formulated using sum of squares but are more complicated and require additional dummy variables.

4.2. Nonparametric Shape-Constrained Regression

Here, the problem is the same as the one discussed in §4.1 with the difference that now we do not have a finite set of parameters θ_j to characterize $f(\mathbf{x})$. Instead, we only assume that $f(\mathbf{x})$ is a continuous and differentiable up to some given order. Technically, we must require that f is in some complete and closed linear space of functions. For example, $f \in S_m(\Delta)$, the Sobolev-Hilbert space. In addition, we require that some linear functional $A(f)$ is nonnegative. In that case, we can use splines of order m with finer and finer grid (or knot sequence in the one-dimensional case) to get better approximations. Of course, now we need to require that the spline is nonnegative over every patch (or interval in the one-dimensional case).

However, as is well known, the problem just stated is not well defined, or the optimal solution produced is not at all satisfactory. For any finite set of input data $(y_1, \mathbf{x}_1), \ldots, (y_n, \mathbf{x}_n)$ one can find an interpolating function in $S_m(\Delta)$; in fact, with a sufficiently fine grid, polynomial splines will do the job. The problem is that an interpolating function is often unsatisfactory in that it is overly dependent on the sample data yet may be a very poor predictor for other values. This phenomenon is known as *overfitting* of data. In addition, if the input data is even moderately large, the interpolating polynomial is very likely to be jagged. To alleviate this problem, it is often advised that a *nonsmoothness penalty* functional be added to the objective function.

Let us first discuss the single variable case in some detail. When $\Delta = [a,b]$ a bounded interval, a common nonsmooth penalty functional is

$$\lambda \int_a^b |f''(x)|^2 \, dx. \tag{43}$$

With this choice of penalty functional, the objective is now to minimize sum of squares of deviations plus the penalty functional: $\sum_i (y_i - f(x_i))^2 + \lambda \int_a^b |f''(x)|^2\, dx$. It can be shown that the minimizer of this penalized least squares objective is a cubic spline. Therefore, as in the parametric case above, we can take the following steps to get a second-order cone program:

• First, we replace the quadratic part $\sum_i (y_i - f(x_i))^2$ with a new variable z_1, and add the SOC constraint

$$z_1^2 \geq \sum_i (y_i - f(x_i))^2$$

to the constraints.

• It is easy to see that in the case of cubic splines, the integral $\int_a^b |f''(x)|^2 dx$ is a positive definite quadratic functional of the coefficients of the spline function $f(\cdot)$. In other words, there is a positive definite matrix R dependent on the knots $\mathbf{t}$ such that

$$\int_a^b \big|f''(x)\big|^2 dx = \mathbf{f}^\top R \mathbf{f}$$

(see de Boor [6]). We can now replace the penalty functional by the variable z_2 and add the SOC constraint

$$z^2 \geq \int_a^b \big|f''(x)\big| dx = \mathbf{f}^\top R \mathbf{f},$$

which is an SOC inequality as discussed in §1.

• To ensure $f(t) \geq 0$ in the interval $[a, b]$, add the constraints in §3.1.2 for each *knot interval* (t_i, t_{i+1}).

The result is an SOCP problem with roughly twice as many SOC inequalities of dimension three as there are knots. This type of problem can be solved relatively efficiently using interior point algorithms; see §5.

For nonnegative multivariate regression, we can use multivariate sum-of-squares splines. If the splines are defined over, for example, a triangular patch, then we can use techniques similar to Example 3.3 for each patch and come up with three times as many semidefinite inequalities as the number of patches. As in the parametric case, this approach can be extended to shape constraints such as convexity by adding additional variables, and replacing *nonnegativity* with *sum of squares*. Study of multivariate convex constraints, even for bivariate functions, is an active area of research.

4.3. Parametric Density Estimation

We are now interested in estimating an unknown (possibly multivariate) density function $f(\mathbf{x})$ with support over a domain $\Delta \subseteq \mathbb{R}^k$. Often, the data are given by a sequence of i.i.d. random variates $\mathbf{x}_1, \ldots, \mathbf{x}_n$ with common density $f(\mathbf{x})$. Our goal is to find the maximum likelihood estimate of the function $f(\mathbf{x})$. In the parametric case, we assume that $f(\mathbf{x}) = \sum_j \theta_j f_j(\mathbf{x})$, which is determined if the parameters θ_j are known. Of course, because $f(\mathbf{x})$ is a density function, it must also satisfy $\int_\Delta f(\mathbf{x}) d\mathbf{x} = 1$ and $f(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \Delta$. The objective in this problem is usually the maximum likelihood functional

$$\prod_{i=1}^n f(\mathbf{x}_i).$$

First, let us take up the univariate case where $\Delta = [a, b]$. If the $f_j(x)$ are assumed to be polynomials, then we use the technique employed by (3) to reduce the objective to a sequence of SOC inequalities. At the end, we will have inequalities of the form $z_i \geq \sum_j \theta_j f_j(x_i)$, which is a linear inequality constraint for each data point x_i. The requirement that

$\int_a^b f(x)dx = 1$ can be expressed again as a linear equality constraint in θ_j. Finally, the nonnegativity constraint can be reduced to semidefinite constraints from §3.1.1. As a result, we obtain a mixed SOCP/SDP problem that can be solved by the interior point method. However, the transformation to SOC inequalities seems to be costly, because we must create n new variables z_i and n SOC inequalities. Instead, we can use the original maximum likelihood objective, or the log-likelihood function $\sum_i \ln f(x_i)$, and apply a more general convex programming algorithm.

By now it should be clear that we may also include additional shape constraints without difficulty. Convexity/concavity, isotonic constraints, and even unimodality (with known mode) can be easily accommodated by semidefinite constraints.

Everything we have said above about density estimation extends to multivariate case. The only issue is the nonnegativity of polynomial $\sum_j \theta_j f_j(\mathbf{x})$, which should be replaced by sum-of-squares condition over Δ.

4.4. Nonparametric Density Estimation

Finally, we consider the same problem as in §4.3, except that now, $f(\mathbf{x})$ is no longer parametrized by a fixed set of parameters θ_j. Instead, we require that $f(\mathbf{x}) \in S_m(\Delta)$. The difficulty is that the solution to the maximum likelihood problem in this case is a linear combinations of Dirac $\delta(\cdot)$ distributions. In other words, the maximum likelihood solution is the "function" that is zero everywhere, except at sample points $\mathbf{x}_i$ on which it is infinite. Even if we attach meaning to such "solutions," the issue of overfitting still remains, and the solution is unusable. To fix the problem, again, a smoothing penalty functional can be added to the maximum likelihood objective function. In this way, we obtain a penalized likelihood function. More precisely, the objective is now to minimize

$$-\frac{1}{n}\sum_j \log f(\mathbf{x}) + \lambda \|f\|^2$$

where $\|f\|$ could be the euclidean norm defined in (41) for the Sobolev-Hilbert space $S_m(\Delta)$. Again, it can be shown that the solution to this problem is a degree m polynomial spline; see Thompson and Tapia [19].

It is possible to get around the smoothness penalty functional by using the method of *cross-validation*. It works as follows: First we fix a particular grid (or simply knot sequence $\mathbf{t}_0$ for the univariate case) and solve the maximum likelihood problem over the space of degree m splines on this space. However, in solving for the most likely spline, we omit a subset of observations $\mathbf{x}_j$. Once the maximum likelihood $f(\mathbf{x})$ spline is determined, we calculate the log-likelihood $\sum_j \ln(f(\mathbf{x}_j))$ over the set of omitted points. Next, we refine the grid by subdividing each patch into smaller patches (in the univariate case we may add more knot points t_j to $\mathbf{t}_0$ to get $\mathbf{t}_1$). We repeat the process mentioned above and again calculate the likelihood of the omitted points as estimated from the remaining (not omitted) points. The process of refining the grid patches continues until the quality of the estimation of the log-likelihood values of the omitted points starts to deteriorate. At that point, overfitting starts to show up and we can stop the refining process.

Let us describe this method for the univariate case with cubic splines. Let the sequence of observations be $x_1, \ldots, x_n$. And let our starting point be the sequence of knots $\mathbf{t}_0 = (t_1, \ldots, t_k)$. If the polynomial on the interval $[t_i, t_{i+1}]$ is $p_i(x)$ and if the sample point x_j falls in the interval $[t_{k_j}, t_{k_j+1}]$, then the log-likelihood function is $-\sum_{i=1}^n \ln(p_{k_i}(x_i))$. The requirement that $\int_a^b f(x)dx = 1$ can be transformed into a linear equality constraint. Finally, the requirement that $f(x) \geq 0$ can be translated into k inequalities $p_j(x) \geq 0$ for all $x \in [t_j, t_{t_j+1}]$. However, from the results of §3.1.2, we know that such inequalities can be expressed as three-dimensional SOC inequalities; our optimization problem has about $2k$ such inequalities.

4.5. A Case Study: Estimation of Arrival Rate of Nonhomogeneous Poisson Process

In Alizadeh et al. [3], we have successfully applied the SDP approach to the problem of estimating the *arrival rate* of a nonhomogeneous Poisson process from observed arrival data. This problem is slightly different from the density estimation in that instead of estimating the density itself, we wish to estimate, nonparametrically, the arrival rate of a Poisson density with time-dependent arrival rate. As an example, consider the arrival of e-mails, visits to a website, customers in a restaurant, or accidents in an intersection. The fundamental assumption is that arrivals are independent of each other; however, the rate of arrival may depend on the time of the day (or date). E-mails may be more frequent during business hours than say Friday nights; customer may enter a restaurant at a faster rate during the lunch hour than say at 10 AM.

The nonhomogeneous Poisson distribution with arrival rate $\lambda(t)$ has the density function

$$\lambda(t)\exp\left(\int_0^t \lambda(t)\right).$$

Clearly, $\lambda(t)$ must be nonnegative. And we will assume that it is smooth and differentiable up to a certain order m; in other words, we assume $\lambda(\cdot) \in S_m([0,T])$.

Our goal is to estimate $\lambda(t)$ from a sequence of observed arrivals $t_1, t_2, \ldots, t_n$. In many practical situations, one may not have *exact* arrival time information, but instead data of the following aggregated form: Given some times $q_0 < q_1 < \cdots < q_k$, we know the number of arrivals n_j in each interval $(q_{j-1}, q_j]$, but not the exact arrival times within these intervals. Here, we can still apply the maximum likelihood principle: an arrival rate function $\lambda\colon [q_0, q_k] \to \mathbb{R}_+$ and the Poisson model assign a probability of

$$P(n_j, q_{j-1}, q_j, \lambda) = \frac{1}{n_j!}\left(\int_{q_{j-1}}^{q_j} \lambda(t)\,dt\right)^{n_j} \exp\left(-\int_{q_{j-1}}^{q_j} \lambda(t)\,dt\right)$$

to the occurrence of n_j arrivals in $(q_{j-1}, q_j]$. Letting $\mathbf{n} = (n_1, \ldots, n_k)$ and $\mathbf{q} = (q_0, \ldots, q_k)$, the joint probability of the arrival pattern $\mathbf{n}$ is

$$P(\mathbf{n}, \mathbf{q}, \lambda) = \prod_{j=1}^{k} P(n_j, q_{j-1}, q_j, \lambda).$$

Again, the maximum likelihood principle suggests choosing $\lambda(\cdot)$ to maximize $P(\mathbf{n}, \mathbf{q}, \lambda)$, or equivalently $L_d(\mathbf{n}, \mathbf{q}, \lambda) = \ln P(\mathbf{n}, \mathbf{q}, \lambda)$. Simplifying L_d, we obtain

$$L_d(\mathbf{n}, \mathbf{q}, \lambda) = \sum_{j=1}^{k}\left(n_j \ln\left(\int_{q_{j-1}}^{q_j} \lambda(t)\,dt\right) - \ln n_j!\right) - \int_{q_0}^{q_k} \lambda(t)\,dt. \tag{44}$$

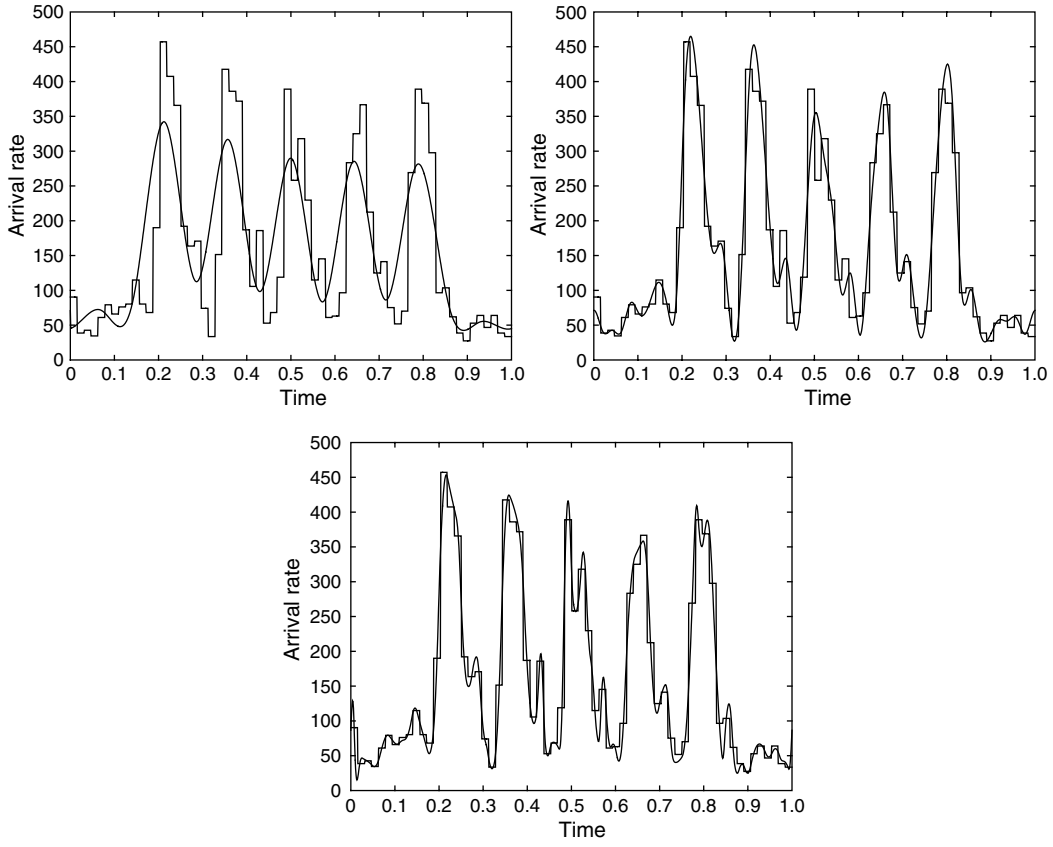
Note that the terms $\ln n_j!$ are independent of λ, and therefore can be ignored when performing the optimization $\max_{\lambda\in\Lambda} L_d(\mathbf{n}, \mathbf{q}, \lambda)$.

We take (44) as our objective function. We represent $\lambda(t)$ by a cubic polynomial spline, with an initially small (equally spaced) knot sequence $\mathbf{t} = (t_0 = 0, t_1, \ldots, t_n = T)$. We use the cross-validation technique, solving subsequent maximum likelihood problems with nonnegativity constraints, until further addition of knots results in overfitting.

This technique was applied to a set of approximately 10,000 e-mails received during a 60-day period. The arrival rate function followed a weekly periodic pattern, which we also incorporated into our optimization model. (The periodicity constraints are expressed by simple linear equality constraints.) The results are shown in Figure 1.

For each of the panels we have shown both the n_j data depicted by a step function and the smooth cubic spline approximation. As can be seen for this particular example, the 14-knot spline is too inaccurate, and the 336 spline overfits the data. Using cross-validation, the best results were achieved around 48-knots.

FIGURE 1. 14-knot, 48-knot, and 336-knot approximation for a large e-mail data set.



5. Interior Point Algorithms

In this section, we will briefly discuss interior point algorithms for solving SDP and SOCP problems. Interior point methods are universal algorithms that are fairly well studied and have predictable behavior. However, these algorithms may not be suitable in certain situations, for instance, when the number of decision variables is extremely large (for example, in the order of tens of thousands) or instances where the "coefficient matrices" A_{ij} are very sparse. On the other hand, interior point algorithms are well suited for the approximation and regression problems where polynomial splines of low degree are used.

To express interior point methods, we first define the notion of a *barrier function*. For a proper cone $\mathcal{K}$, a function $b(\mathbf{x})$ is a barrier function if

(1) b: $\operatorname{Int}\mathcal{K} \to \mathbb{R}$ is a convex function.

(2) For any sequence of points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k, \ldots$ where $\mathbf{x}_k$ converges to a point in the boundary of $\mathcal{K}$ as $k \to \infty$ the value of the barrier $b(\mathbf{x}_k) \to \infty$.

To see how barrier functions are used, consider primal problem in (28) but add the barrier to the objective function:

$$\begin{aligned} \min \quad & \mathbf{c}^\top \mathbf{x} + \mu b(\mathbf{x}) \\ \text{s.t.} \quad & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \in \mathcal{K}. \end{aligned} \tag{45}$$

When the parameter μ is large, the term $\mu b(\mathbf{x})$ is dominant in the objective function. And because $b(\mathbf{x})$ is infinite at the boundary of cone $\mathcal{K}$, the minimum is attained at $\mathbf{x}_\mu$, a point

well in the interior of $\mathcal{K}$. On the other hand, if μ is small, $\mathbf{c}^\top \mathbf{x}$ is the dominant term, and the solution $\mathbf{x}_\mu$, while still in the interior of $\mathcal{K}$, is nonetheless close to the minimizer of primal problem (28). The overall strategy of interior point methods now emerges. We start with μ_0 relatively large. It turns out that under some general conditions, (45) is well behaved numerically and can be solved without much difficulty. Next, iteratively, we reduce μ_0 by a factor to get μ_1, and use the previous optimal $\mathbf{x}_{\mu_0}$ as the initial estimate of (45) with parameter μ_1. We solve the new optimization problem to get $\mathbf{x}_{\mu_1}$. Again, we reduce μ_1 by some factor to get μ_2. This process is continued until μ_k is sufficiently small, and thus $\mathbf{x}_{\mu_k}$ is close to the optimal solution $\mathbf{x}^*$ of (28). The main problem to solve in this general scheme is to determine by what factor we should reduce μ_k to μ_{k+1} so that

(1) $\mathbf{x}_{\mu_k}$ is fairly close to $\mathbf{x}_{\mu_{k+1}}$, making computational effort of finding $\mathbf{x}_{\mu_{k+1}}$ starting from $\mathbf{x}_{\mu_k}$ not too expensive, and

(2) μ_{k+1}/μ_k is fairly large, so that the sequence μ_k converges to zero rather quickly, thereby making the sequence $\mathbf{c}^\top \mathbf{x}_{\mu_k}$ converge to the optimal solution $\mathbf{c}^\top \mathbf{x}^*$ quickly.

Note that the two criteria above are opposites of each other. In many variants of interior point methods, it is expected that only one—or at most very few—iterations are required to find $\mathbf{x}_{\mu_{k+1}}$ from $\mathbf{x}_{\mu_k}$.

5.1. Interior Point Methods for Semidefinite Programming

We now discuss the class of *primal-dual* interior point methods for SDP. First, it is fairly easy to prove that for the semidefinite cone the function $-\ln \operatorname{Det} X$ is a barrier. We will deal with the case in which we have only one matrix variable. First, we replace the primal SDP with

$$\begin{aligned} \min \quad & C \bullet X - \mu \ln \operatorname{Det} X \\ \text{s.t.} \quad & A_i \bullet X = b_i. \end{aligned} \tag{46}$$

Next, we write the Lagrangian function

$$L(X, y) = C \bullet X - \mu \ln \operatorname{Det} X - \sum_i y_i (b_i - A_i \bullet X)$$

where the y_i are the Lagrange multipliers. The optimality conditions now imply that X_μ is optimal for (46) if there is $\mathbf{y}_\mu$ such that

$$\nabla_X L(X, \mathbf{y}) = C - \mu X^{-1} - \sum_i y_i A_i = 0 \tag{47}$$

$$\nabla_{\mathbf{y}} L(X, \mathbf{y}) = (b_i - A_i \bullet X)_{i=1}^m = \mathbf{0}. \tag{48}$$

A few words are in order. First, because X is a symmetric matrix, the gradient $\nabla_X L$ is a matrix-valued functional. Second, the gradient of $\ln \operatorname{Det} X$ is X^{-1}. Third, the gradient $\nabla_{\mathbf{y}} L$ is a vector of size m whose ith entry is $b_i - A_i \bullet X$. Finally, observe that if $X \succ 0$, then $X^{-1} \succ 0$ as well. Thus, (47) indicates that the matrix $S = \mu X^{-1}$ is dual feasible and, indeed, in the interior of the positive semidefinite cone. It follows that $XS = \mu I$ or equivalently $(XS + SX)/2 = \mu I$. Therefore, (47) and (48) can be combined to produce the system of equations

$$\begin{aligned} & A_i \bullet X = b_i \text{ for } i = 1, \ldots, m \\ & \sum_i y_i A_i - S = C \\ & \frac{XS + SX}{2} = \mu I. \end{aligned} \tag{49}$$

Observe that this system includes primal feasibility, dual feasibility, and a relaxed form of complementarity condition for SDP. In fact, if we set $\mu = 0$, we obtain exactly the

complementary conditions. Assuming that we have an initial primal-dual feasible solution $(X_0, \mathbf{y}_0, S_0)$ that solves (49) for $\mu = \mu_0$. We can apply *Newton's* method to iteratively generate a sequence of primal-dual points $(X_k, \mathbf{y}_k, S_k)$, which converge to the optimum $(X^*, \mathbf{y}^*, S^*)$ of the primal-dual SDP problem. Applying Newton's method involves replacing $(X, \mathbf{y}, S)$ in (46) with $(X + \Delta X, \mathbf{y} + \Delta\mathbf{y}, S + \Delta S)$, rearranging the resulting set of equation in terms of $(\Delta X, \Delta\mathbf{y}, \Delta S)$, removing all nonlinear terms in Δ's, and solving the resulting linear system of equations for Δ's. Carrying out this procedure, we get

$$\begin{aligned} & A_i \bullet \Delta X = b_i - A_i \bullet X \\ & \textstyle\sum_i \Delta y_i A_i + \Delta S = C - \sum_i y_i A_i \\ & X\Delta S + \Delta S X + S\Delta X + \Delta X S = 2\mu I - (XS + SX) \\ \Leftrightarrow \quad & \begin{pmatrix} \mathcal{A} & 0 & 0 \\ 0 & \mathcal{A}^\top & I \\ \mathcal{S} & 0 & \mathcal{X} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{X} \\ \Delta\mathbf{y} \\ \Delta\mathbf{S} \end{pmatrix} = \begin{pmatrix} \delta\mathbf{X} \\ \delta\mathbf{y} \\ \delta\mathbf{S} \end{pmatrix}, \end{aligned} \tag{50}$$

where $\mathcal{A}$ is the linear transformation sending X to $\mathbf{b}$, and $\delta\mathbf{X}, \delta\mathbf{y}, \delta\mathbf{S}$ are the right side of the system. Finally, $\mathcal{X}$ and $\mathcal{S}$ are matrices that are linearly dependent on X and S.

This system of equations can be solved for Δ's and yields the Newton direction. Typical interior point methods may apply some scaling of the matrix $\mathcal{A}$ to get systems with more favorable numerical properties. Once this system is solved, a new interior point $(X + \alpha_k \Delta X, \mathbf{y} + \beta_k \Delta\mathbf{y}, S + \gamma_k \Delta S)$ emerges. The process is repeated by reducing μ until we are sufficiently close to the optimal solution. Notice that both feasibility of the solution and its optimality can be gauged at each point: The size of $(b_i - A_i \bullet X_k)$, $C - \sum_i (\mathbf{y}_k)_i A_i - S_k$, indicate primal and dual infeasibility, and $X_k \bullet S_k$ indicate the duality gap. With judicious choice of step lengths $\alpha_k, \beta_k, \gamma_k$ and a reduction schedule μ_{k+1}/μ_k, it is possible to design an efficient and fast-converging algorithm.

5.2. Interior Point Methods for SOCP

For second-order cone $\mathcal{Q}$, the function $\ln(x_0^2 - \|\bar{\mathbf{x}}\|^2)$ is a barrier. Following the same procedure as in SDP (and working only with one block of variables for ease of presentation), we replace the primal second-order cone program with

$$\begin{aligned} \min \quad & \mathbf{c}^\top \mathbf{x} - \mu \ln(x_0^2 - \|\bar{\mathbf{x}}\|^2) \\ \text{s.t.} \quad & A\mathbf{x} = \mathbf{b}. \end{aligned} \tag{51}$$

With Lagrange multiplier $\mathbf{y}$, the Lagrangian is given by

$$L(\mathbf{x}, \mathbf{y}) = \mathbf{c}^\top \mathbf{x} - \ln(x_0^2 - \|\bar{\mathbf{x}}\|^2) + \mathbf{y}^\top (\mathbf{b} - A\mathbf{x}).$$

Applying the standard optimality conditions gives

$$\begin{aligned} \nabla_{\mathbf{x}} L = \mathbf{c}^\top - \frac{2\mu}{x_0^2 - \|\bar{\mathbf{x}}\|^2}(x_0, -x_1, \ldots, -x_n) - y^\top A = \mathbf{0}^\top \\ \mathbf{b} - A\mathbf{x} = \mathbf{0}. \end{aligned}$$

Define $\mathbf{s}^\top = (2\mu/(x_0^2 - \|\bar{\mathbf{x}}\|^2))(x_0, -x_1, \ldots, -x_n)$. Then, obviously, $\mathbf{x} \in \operatorname{Int} \mathcal{Q}$ if and only if $\mathbf{s} \in \operatorname{Int} \mathcal{Q}$. Thus, $\mathbf{s}$ is dual feasible and in the interior of $\mathcal{Q}$. It can be shown that $\mathbf{s}$ is, in fact, the unique vector satisfying

$$\mathbf{x}^\top \mathbf{s} = \mu \quad \text{and} \quad x_0 s_i + s_0 x_i = 0.$$

Thus, the optimality conditions can be written as

$$\begin{aligned} &A\mathbf{x} = \mathbf{b} \\ &A^\top \mathbf{y} + \mathbf{s} = \mathbf{c} \\ &\mathbf{x}^\top \mathbf{s} = 2\mu \\ &x_0 s_i + s_0 x_i = 0 \quad \text{for } i = 1, \ldots, n. \end{aligned} \tag{52}$$

Observe that the last two sets of equations are relaxations of the complementarity slackness relations for SOCP. Thus, again, as μ tends to zero, the solution $(\mathbf{x}_\mu, \mathbf{y}_\mu, \mathbf{s}_\mu)$ tends to the optimal solution of SOCP. As in the case of SDP, we can solve (52) by applying Newton's method. We replace $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ with $(\mathbf{x} + \Delta\mathbf{x}, \mathbf{y} + \Delta\mathbf{y}, \mathbf{s} + \Delta\mathbf{s})$, and remove all terms nonlinear in Δ's to arrive at the system

$$\begin{aligned} &A\Delta\mathbf{x} = \mathbf{b} - A\mathbf{x} \\ &A^\top \Delta\mathbf{y} + \Delta\mathbf{s} = \mathbf{c} - A^\top \mathbf{y} - \mathbf{s} \\ &\mathbf{x}^\top \Delta\mathbf{s} + \mathbf{s}^\top \Delta\mathbf{x} = mu - \mathbf{x}^\top \mathbf{s} \\ &x_0 \delta s_i + \delta x_0 s_i + s_0 \Delta x_i + x_i \Delta s_i = -x_0 s_i - s_0 x_i \end{aligned} \quad \Longleftrightarrow \quad \begin{pmatrix} A & 0 & 0 \\ 0 & A^\top & I \\ \operatorname{Arw} \mathbf{s} & 0 & \operatorname{Arw} \mathbf{x} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{s} \\ \Delta\mathbf{y} \\ \Delta\mathbf{s} \end{pmatrix} = \begin{pmatrix} \delta\mathbf{x} \\ \delta\mathbf{y} \\ \delta\mathbf{s} \end{pmatrix}$$

where

$$\operatorname{Arw} \mathbf{x} = \begin{pmatrix} x_0 & \bar{\mathbf{x}}^\top \\ \bar{\mathbf{x}} & x_0 I \end{pmatrix}.$$

and $(\delta\mathbf{x}, \delta\mathbf{y}, \delta\mathbf{s})$ are the right-hand side of the system.

Similar to SDP, one starts with a given solution $(\mathbf{x}_0, \mathbf{y}_0, \mathbf{s}_0)$ that is an estimate of (52). After solving for the Δ's, a new estimate $(\mathbf{x} + \alpha_k \Delta\mathbf{x}, \mathbf{y} + \beta_k \Delta\mathbf{y}, \mathbf{s} + \gamma_k \Delta\mathbf{s})$ is computed and μ is reduced by a factor. With judicious choice of step lengths $\alpha_k, \beta_k, \gamma_k$ and a reductions schedule for μ, we can get fast-converging interior point algorithm.

5.3. Available SDP and SOCP Software

Variants of interior point methods as discussed in the previous two sections are implemented in several open-source packages. Currently, the most popular package for solving both SDP and SOCP problems is a package developed by late Jos Sturm called SeDuMi Sturm [18]. This package is written in Matlab, though most of its critical inner code is in C. It is based on a variant of primal-dual interior point known as the Nesterov-Todd method [14, 15]. The software is designed to be numerically very stable.

Other software include SDPpack of Alizadeh et al. [4], SDPA of Fujisawa et al. [9], and SDPT3 of Tutuncu et al. [20]. All of these packages are Matlab based, freely available, and open-source. The main drawback of them all is that they require both linear objective and linear functionals on both sides of SDP and SOC inequality constraints. This makes such software hard to use for situations in which the objective function is nonlinear, for example, as in the case of log-likelihood functions.

An alternative is using general-purpose nonlinear programming software. Two of the most successful ones are KNITRO of Nocedal and Waltz [16] and LOQO of Vanderbei [22]. These packages are commercial, and their source code is not freely available. They are, however, useful for small to medium-size second-order cone programs with possibly nonlinear objective function. In fact, the case study discussed in §4.5 was solved using KNITRO. Unfortunately, these packages do not have effective means of handling semidefinite constraints.

To our knowledge, there is currently no polished, public package—commercial or open-source—that can handle nonlinear optimization problems with nonlinear semidefinite objective and linear SDP or SOCP constraints. There is no particular difficulty in writing such code, at least when the objective is convex (or concave in the case of maximization problems).

6. Concluding Remarks

This survey represents only an introduction to the theory and applications of SDP. Use of SDP in shape-constrained approximation and regression discussed here is fairly new and the subject of active current research. Of particular interest are the case of shape-constrained multivariate regression and estimation.

Other applications, as well as more thorough study of the theory and algorithms, are discussed in the collection of papers by Saigal et al. [17] for SDP and the survey article of Alizadeh and Goldfarb [2] for the SOCP.

Acknowledgments

The author would like to thank Michael Johnson for making helpful suggestions that improved the presentation. Research supported in part by U.S. National Science Foundation Grant NSF-CCR-0306558 and Office of Naval Research through Contract N00014-03-1-0042.

References

[1] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization* 5(1):13–51, 1995.

[2] F. Alizadeh and D. Goldfarb. Second-order cone programming. *Mathematical Programming Series B* 95:3–51, 2003.

[3] F. Alizadeh, J. Eckstein, N. Noyan, and G. Rudolf. Arrival rate approximation by nonnegative cubic splines. Technical Report RRR 46-2004, RUTCOR, Rutgers University, Piscataway, NJ, 2004.

[4] F. Alizadeh, J. P. A. Haeberly, V. Nayakkankuppam, M. L. Overton, and S. A. Schmieta. SDPPACK user guide, version 0.9 beta. Technical Report 737, Courant Institute of Mathematical Sciences, New York University, New York, 1997. http://www.cs.nyu.edu/faculty/overton/sdppack.

[5] C. K. Chui. Multivariate splines. *CBMS-NSF*, Vol. 54. SIAM, Philadelphia, PA, 1988.

[6] C. de Boor. *A Practical Guide to Splines*. Springer-Verlag, New York, 1978.

[7] H. Dette and W. J. Studden. *The Theory of Canonical Moments with Applications in Statistics, Probability, and Analysis*. Wiley Interscience Publishers, New York, 1997.

[8] L. Faybusovich. Self-concordant barrieres for cones generated by Chebyshev systems. *SIAM Journal on Optimization* 12(3):770–781, 2002.

[9] K. Fujisawa, M. Kojima, K Nakata, and M. Yamashita. SDPA (semidefinite programming algorithm) user's manual, version 6.2.0. Technical Report B-308, Department of Mathematics and Computer Sciences, Tokyo Institute of Technology, 2004.

[10] S. Karlin and W. J. Studden. *Tchebycheff Systems, with Applications in Analysis and Statistics*. Wiley Interscience Publishers, New York, 1966.

[11] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second order cone programming. *Linear Algebra Applications* 284:193–228, 1998.

[12] Y. Nesterov. Squared functional systems and optimization problems. J. B. G. Frenk, C. Roos, T. Terlaky, and S. Zhang, eds. *High Performance Optimization*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 405–440, 2000.

[13] Y. Nesterov and A. Nemirovski. *Interior Point Polynomial Methods in Convex Programming: Theory and Applications*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994.

[14] Y. E. Nesterov and M. J. Todd. Self-scaled barriers and interior-point methods for convex programming. *Mathematics of Operation Research* 22:1–42, 1997.

[15] Y. E. Nesterov and M. J. Todd. Primal-dual interior-point methods for self-scaled cones. *SIAM Journal on Optimization* 8:324–364, 1998.

[16] J. Nocedal and R. A. Waltz. KNITRO user's manual. Technical Report OTC 2003/05, Northwestern University, Evanston, IL, 2003.

[17] R. Saigal, L. Vandenberghe, and H. Wolkowicz, eds. *Handbook of Semidefinite Programming, Theory, Algorithms, and Applications*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.

[18] J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software* 11–12:625–653, 1999.

[19] J. R. Thompson and R. A. Tapia. *Nonparametric Function Estimation, Modeling, and Simulation*. SIAM, Philadelphia, PA, 1990.

[20] R. H. Tutuncu, K. C. Toh, and M. J. Todd. SDPT3—A Matlab software package for semidefinite-quadratic-linear programming, version 3.0. Technical report, Department of Mathematics, National University of Singapore, Singapore, 2001.

[21] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review* 38(1):49–95, 1996.

[22] R. J. Vanderbei. LOQO user's manual. Technical Report ORFE-99, Operations Research and Financial Engineering, Princeton University, Princeton, NJ, 2000.

[23] G. Wahba. *Spline Models for Observational Data*. SIAM, Philadelphia, PA, 1990.

TUTORIALS IN
OPERATIONS RESEARCH
INFORMS 2006

 | ISBN 13 978-1-877640-20-9
DOI 10.1287/educ.1063.0021

Model Uncertainty, Robust Optimization, and Learning

Andrew E. B. Lim, J. George Shanthikumar, and Z. J. Max Shen
Department of Industrial Engineering and Operations Research,
University of California, Berkeley, California 94720
{lim@ieor.berkeley.edu, shanthikumar@ieor.berkeley.edu, shen@ieor.berkeley.edu}

Abstract Classical modeling approaches in OR/MS under uncertainty assume a full probabilistic characterization. The learning needed to implement the policies derived from these models is accomplished either through (i) *classical statistical estimation procedures* or (ii) *subjective Bayesian priors*. When the data available for learning is limited, or the underlying uncertainty is nonstationary, the error induced by these approaches can be significant and the effectiveness of the policies derived will be reduced. In this tutorial, we discuss how we may incorporate these errors in the model (that is, model *model uncertainty*) and use *robust optimization* to derive efficient policies. Different models of model uncertainty will be discussed and different approaches to *robust optimization with and without benchmarking* will be presented. Two alternative learning approaches—*objective Bayesian learning* and *operational learning*—will be discussed. These approaches could be used to calibrate the models of model uncertainty and to calibrate the optimal policies. Throughout this tutorial, we will consider the classical inventory-control problem, the inventory-control problem with censored demand data, and the portfolio-selection problem as examples to illustrate these ideas.

Keywords model uncertainty; robust optimization; learning; operational statistics

1. Introduction

The majority of the early models in OR/MS have been deterministic. Specifically, models for production planning, logistics, and transportation have been based on the assumption that all variables of interest are known in advance of the implementation of the solutions. While some models, such as queueing, insurance, and portfolio selections naturally call for incorporating stochasticity, it is usually assumed that the full probabilistic characterization of these models are known in advance of the implementation of the solutions. Even when it is assumed that the parameters of a parametric stochastic model are unknown, it is assumed that a Bayesian prior for the parameters is known (e.g., Azoury [10], Berger [15], Ding et al. [39], Robert [82]). Such an approach is often justified by the axiomatic framework of Savage [84] for decision making—assuming this one ends up with a model that has been fully characterized. In economics, with the initial work of Knight [70] and the Ellsberg paradox [43], questions on this basic idea of full probabilistic characterization have been raised. The seminal work of Gilboa and Schmeidler [57] provides an axiomatic framework justifying the notion of multiple fully characterized stochastic models for a single decision problem with a max-min objective. This sparked the basis for model uncertainty and robust optimization in the economics and finance areas (e.g., Anderson et al. [3, 4], Cagetti et al. [28], Cao et al. [29], Dow and Werlang [40], Epstein [44], Epstein and Miao [45], Epstein and Schneider [47, 48, 49], Epstein and Wang [50], Garlappi et al. [56], Hansen and Sargent [59, 60, 61]). For a recent account of the application of model uncertainty and robust optimization in economics

and finance, see the monograph by Hansen and Sargent [62]. Within the OR/MS, community interest in deterministic robust optimization has been strong recently (e.g., Atamturk [5], Atamturk and Zhang [6], Averbakh [7, 8, 9], Ben-Tal and Nemirovski [11, 12, 13, 14], Bertsimas and Sim [20, 21, 22], Bertsimas et al. [24], El Ghaoui and Lebret [41], El Ghaoui et al. [42]). See Soyster [86] for one of the earliest contributions to this area and the book by Kouvelis and Yu [71] for a detailed account of the developments until the mid '90s. However, stochastic models of model uncertainty have not received as much attention as the others in the OR/MS literature. In this tutorial, we will describe the different ideas in modeling model uncertainty, finding the solution to this model using robust optimization, and its implementation through learning.

Consider a static or a discrete time dynamic optimization problem defined on a sample space $(\Omega, \mathcal{F}, (\mathcal{F}_k)_{k\in\mathcal{M}})$. Here, $\mathcal{M} = \{0, 1, 2, \ldots, m\}$, where m is the number of decision epochs ($m = 1$ for a static optimization problem, $m = 2$ in a stochastic programming problem with recourse, and $m \geq 2$ for a discrete dynamic optimization problem). Ω is the set of all possible outcomes of the input variables Y_0 and the future values $\mathbf{Y} = \{Y_k, k = 1, 2, \ldots, m\}$ of interest for the optimization problem (such as the demand over time for different items in an inventory-control problem, the arc lengths and costs in a network optimization problem, etc.). $\mathcal{F}$ is the sigma algebra of event in Ω, and $\mathcal{F}_0$ is (the sigma algebra of) all possible information on the input variables that may be available to the decision maker at time 0 (such as the past demand or sales data for the different items in an inventory-control problem or the arc lengths and costs in network optimization problem). The actual information $\mathcal{I}_0$ available to the decision maker is an element of $\mathcal{F}_0$. Though it is not required, $\mathcal{F}_n$ is often the sigma algebra generated by the internal history of the variables $\{Y_k, k \in \mathcal{M}\}$ (that is, $\mathcal{F}_k = \sigma(Y_j, j = 0, 1, 2, \ldots, k)$). It should be noted that the information available to the decision maker at the beginning of period $k + 1$ ($k \geq 1$) may not be $\mathcal{F}_k$ (for example, in an inventory-control problem, one may only have information on the sales and not the actual demand values).

Let π_1 be the decision made at the beginning of Period 1 (which is adapted to an information subset $\mathcal{I}_0$ in $\mathcal{F}_0$). This leads to an information set that may depend on π_1. Let $\mathcal{I}_1(\pi_1)$ be the sigma algebra generated by this information set (which satisfies $\mathcal{I}_1(\pi_1) \subset \mathcal{F}_1$). Now, let π_2 be the decision made at the beginning of Period 2 (which is adapted to $\mathcal{I}_1(\pi_1)$). In general, the policy π is adapted to an information filtration $((\mathcal{I}_k(\pi))_{k\in\mathcal{M}})$, which, in turn, is sequentially generated by the policy π.

Let $\psi(\pi, \mathbf{Y})$ be the reward obtained with policy π and Γ be the collection of all admissible policies π. We are then interested in finding a policy $\pi^* \in \Gamma$ that maximizes $\psi(\pi, \mathbf{Y})$ in some sense. One may adapt several alternative approaches to do this. All approaches in some way need to define a probability measure (say P) on $(\Omega, \mathcal{F}, (\mathcal{F}_k)_{k\in\mathcal{M}})$ given $\mathcal{I}_0$. Classical modeling approaches in OR/MS under uncertainty assume that a full probabilistic characterization can be done very accurately (that is, we have *perfect forecasting capability* when a nondegenerate measure is used in our model and that we have the capability to *predict the future perfectly* when the assumed measure is degenerate). When we do this, we hope one or both of the following, assumptions is true.

Assumption (A1). The chosen probability measure P is the true probability measure P_0 or very close (in some sense) to it.

Assumption (A2). The solution (optimal in some sense) obtained with P leads to a performance that is either optimal or close to optimal (in some sense) with respect to P_0.

The learning needed to implement the policies derived from these models is accomplished either through (i) *classical statistical estimation procedures* or (ii) *subjective Bayesian priors*. It is not hard to see that the assumptions in many cases need not be true. When the data available for learning is limited, or the underlying uncertainty is nonstationary, the error induced by these approaches can be significant and the effectiveness of the policy derived will be reduced. In this tutorial, we discuss how we may incorporate these errors in the model (that is, model *model uncertainty*) and use *robust optimization* to derive efficient policies.

Different models of model uncertainty will be discussed, and different approaches to *robust optimization with and without benchmarking* will be presented. Two alternative learning approaches—*objective Bayesian learning* and *operational Learning*—will be discussed. These approaches could be used to calibrate the models of model uncertainty and obtain robust optimal policies.

Before proceeding further with this discussion, we will introduce a very simple canonical example: *The newsvendor inventory problem with demand observed*. This can be thought of as a sequence of n static problems. This model is almost always used as a rat to experiment with to test different ideas in inventory control. It will allow us to discuss the importance of model uncertainty and the integration of optimization and estimation. Later, in §7, we will work out three classes of dynamic optimization problems that will serve as examples to illustrate our ideas on learning with integrated dynamic optimization and estimation and robust optimization with benchmarking.

The Inventory Rat. Consider the perishable item inventory-control problem. Items are purchased at c per unit and sold for s per unit. There is no salvage value and no lost sales penalty. Suppose $Y_1, Y_2, \ldots, Y_m$ represent the demand for this item for the next m periods. We wish to find the optimal order quantities for the next m periods. Suppose we order π_k units in period k. Then, the profit is

$$\psi(\pi, \mathbf{Y}) = \sum_{k=1}^{m} \{s \min\{Y_k, \pi_k\} - c\pi_k\}.$$

This problem allows us to illustrate the effects of separating modeling and optimization from model calibration without having to bring in the consequences of cost-to-go (that is, residual) effects of current decisions at each decision epoch on future time periods. In evaluating the different approaches, we will assume that $Y_1, Y_2, \ldots, Y_m$ are i.i.d. with an absolutely continuous distribution function F_Y. Further, if needed, we will assume that Y_k is exponentially distributed with mean θ (that is, $F_Y(y) = 1 - \exp\{-(1/\theta)y\}$, $y \geq 0$). Let $\{X_1, X_2, \ldots, X_n\}$ be the past demand for the last n periods. This information is contained in Y_0. We will also assume that $\{X_1, \ldots, X_n\}$ are i.i.d. samples from the same distribution as Y_k.

In §2, we will discuss what is done now: How models are formulated, optimized, and implemented. Following a discussion on the possible errors in the current approaches in §2, alternative approaches to model these errors through flexible modeling will be discussed in §3. Flexible modeling will be accomplished through defining a collection of models that is very likely to contain the correct model or a close approximation of it. Hence, finding a robust solution to these model collections depends on defining a robust optimization approach. Alternative approaches to robust optimization are discussed is §4. Section 5 is devoted to the calibration of flexible models using classical statistics. Integrated learning in flexible models using (i) min-max, duality, and objective Bayesian learning, and (ii) operational learning is introduced in §6. Detailed application of the concepts discussed in this tutorial to dynamic inventory-control and portfolio selection are given in §7.

2. Modeling, Optimization, and Implementation

Almost always, the abstract formulation of the model and optimization is done independent of $\mathcal{I}_0$ and how the model will be calibrated. Here, and in the remaining of the paper, we will assume that Y_0 contains the past n values $\{X_k, k = 1, 2, \ldots, n\}$ that will be used to calibrate $\mathbf{Y}$ (that is, its probability measure P).

2.1. Deterministic Modeling, Optimization, and Implementation

Though this is obvious, we wish to discuss deterministic modeling here because it forms a basis for a large body of work currently being done in robust optimization (see the special

issue of *Mathematical Programming*, **107**(1–2), on this topic). Let $P^d_{\omega_0} = I\{\omega = \omega_0\}$, $\omega_0 \in \Omega$ be a collection of degenerate (Dirac) probability measures on $(\Omega, \mathcal{F}, (\mathcal{F}_k)_{k\in\mathcal{M}})$. In deterministic modeling, one assumes that for some chosen $\omega_0 \in \Omega$, we have $P = P^d_{\omega_0}$. Then

$$\phi(\pi, \omega_0) = E[\psi(\pi, \mathbf{Y})] = \psi(\pi, \mathbf{Y}(\omega_0)).$$

Given that the feasible region of π is Γ, one then has the following optimization problem:

$$\phi^d(\omega_0) = \max_{\pi\in\Gamma}\{\phi(\pi, \omega_0)\},$$

and choose a $\pi^d(\omega_0) \in \Gamma$ such that

$$\phi(\pi^d(\omega_0), \omega_0) = \phi^d(\omega_0).$$

To implement this policy, however, one would have to estimate $\mathbf{Y}(\omega_0)$. For example, one may assume that $\{X_1, \ldots, X_n, Y_1, \ldots, Y_m\}$ are i.i.d. and estimate $\mathbf{Y}(\omega_0)$ by, say,

$$\hat{Y}_k(\omega_0) = \bar{X}, \quad k = 1, 2, \ldots, m,$$

where

$$\bar{X} = \frac{1}{n}\sum_{k=1}^{n} X_k.$$

For some problems, the effect of variability on the final solution may be insignificant so that such an assumption of *determinism* can be justified. For most real problems, however, such an assumption may be unacceptable. Often, such an assumption is made so that the resulting optimization problems are linear programs or integer linear programs so that some of the well-established approaches in OR can be used to solve these optimization problems. Sometimes, even with this assumption of determinism, the solution may be hard to get. It is fair to say that the decision to assume determinism is mostly motivated by the desire to get a solution rather than to capture reality. However, with all the advances that have been made in convex optimization (e.g., Bertsekas [18], Boyd and Vandenberghe [27]) and in stochastic programming (e.g., Birge and Louveaux [26], Ruszczynski and Shapiro [83], van der Vlerk [89]), it seems possible to relax this assumption and proceed to formulate stochastic models. Before we proceed to discuss stochastic modeling, we will give the deterministic version of the inventory rat. We will later use this result in robust optimization with benchmarking.

The Inventory Rat (cont'd.).

$$\phi^d(\omega_0) = \max\left\{\sum_{k=1}^{m} \psi(\pi_k, Y_k(\omega_0)):\ \pi_k \geq 0\right\} = (s-c)\sum_{k=1}^{m} Y_k(\omega_0)$$

and

$$\pi^d_k(\omega_0) = Y_k(\omega_0), \quad k = 1, 2, \ldots, m.$$

Then, the expected profit is

$$\phi^d(\theta) = (s-c)m\theta.$$

where $\theta = E[Y_k]$.

To implement this policy, we need to know the future demand. If we do not, maybe we can approximate the future demand by the observed average. Hence, the *implemented policy* would be

$$\hat{\pi}^d_k = \bar{X}, \quad k = 1, 2, \ldots, m$$

with profit

$$\hat{\psi}(Y) = \sum_{k=1}^{m}\{s\min\{Y_k, \bar{X}\} - c\bar{X}\},$$

where $\bar{X} = (1/n)\sum_{k=1}^{n} X_k$. Depending on when policy change is allowed, reoptimization will take place in the future. Here, and in the rest of the paper, we will assume that we are allowed to reoptimize at the end of each period. Now, depending on the belief we have on the i.i.d. assumption for the demand, we may be willing to estimate the demand for the next period based only on the last, say, l periods. For ease of exposition, we will assume that $l = n$. Set $X_{n+j} = Y_j$, $j = 1, 2, \ldots, m$. Then, using an updated estimate of $Y_k(\omega_0)$ at the beginning of period k, we get

$$\hat{\pi}_k^d = \bar{X}_k, \quad k = 1, 2, \ldots, m,$$

where $\bar{X}_k = (1/n)\sum_{j=k}^{n+k-1} X_j$ is the n-period moving average for $k = 1, 2, \ldots, m$. The associated profit is

$$\hat{\psi}(Y) = \sum_{k=1}^{m} \{s \min\{Y_k, \bar{X}_k\} - c\bar{X}_k\}.$$

Suppose the demand is exponentially distributed with mean θ. It is easy to verify that

$$\lim_{m\to\infty} \frac{1}{m}\hat{\psi}(Y) = (s-c)\theta - s\theta\left(\frac{n}{n+1}\right)^n.$$

As $n \to \infty$, one gets an average profit of $(s-c)\theta - s\theta \exp\{-1\}$. It can be verified that this profit can be very inferior to the optimal profit. For example, when $s/c = 1.2$, $c = 1$, and $\theta = 1$, the optimal profit is 0.121 while the above policy results in a profit of -0.241.

2.2. Stochastic Modeling and Optimization

For stochastic modeling, we assume a nondegenerate probability measure. That is, we define, given $\mathcal{I}_0$ a nondegenerate probability measure P on $(\Omega, \mathcal{F}, (\mathcal{F}_k)_{k\in\mathcal{M}})$. Wanting to specify a probability measure without any statistical assumption is indeed an idealized goal. Even if we are able to solve the resulting optimization problem, the calibration of P given $\mathcal{I}_0$ will almost always require us to make some statistical assumptions regarding $\mathbf{Y}$ and Y_0. These assumptions are often such as i.i.d., Markovian, autoregressive of some order, etc. If the state space of $\mathbf{Y}$ is finite, then we may try to solve the problem with respect to the probabilities assigned to the different states (treating them as parameters). Even then, it may be difficult to solve the optimization problem. In such cases and in cases where further information on the distributional characteristic are known, we make additional assumptions that allow one to fully characterize P up to some finite dimensional parameter.

2.2.1. Parametric Modeling, Optimization, and Implementation. Suppose we have fully characterized P up to some finite dimensional parameter, say, θ. For example, this may be achieved by postulating that Y_k has an exponential or normal distribution or that the transition kernel of the Markov process $\mathbf{Y}$ is parameterized by a finite set or the state space if finite. Let P_θ^p be the corresponding probability measure parameterized by θ. Define

$$\phi^p(\pi, \theta) = E[\psi(\pi, \mathbf{Y})].$$

Finding the solution to this formulation depends on one of two approaches one chooses for implementation: frequentist or Bayesian approach.

Frequentist Approach. Suppose we assume that the information $\mathcal{I}_0$ we have will allow us to estimate the parameter θ *exactly.* Then one solves

$$\phi^p(\theta) = \max_{\pi\in\Gamma}\{\phi(\pi, \theta)\},$$

and choose a $\pi^p(\theta) \in \Gamma$ such that

$$\phi(\pi^p(\theta), \theta) = \phi^p(\theta).$$

To implement this policy, however, one would have to estimate θ. Suppose we use some statistical estimator $\hat{\Theta}(\mathbf{X})$ of θ using the data $\mathbf{X}$. Then, we would implement the policy

$$\hat{\pi}^p = \pi^p(\hat{\Theta}(\mathbf{X})).$$

The Inventory Rat (cont'd.). When the demand is exponentially distributed, one has (e.g., Liyanage and Shanthikumar [80], Porteus [81], Zipkin [91]),

$$\phi^p(\pi,\theta) = E[\psi(\pi,\mathbf{Y})] = s\theta\left(1-\exp\left\{-\frac{\pi}{\theta}\right\}\right) - c\pi,$$
$$\pi^p(\theta) = \theta\ln\left(\frac{s}{c}\right),$$

and

$$\phi^p(\theta) = (s-c)\theta - c\theta\ln\left(\frac{s}{c}\right).$$

For an exponential distribution, the sample mean is the uniformly minimum variance unbiased (UMVU) estimator. Hence, we will use the sample mean of the observed data to estimate θ. Then the *implemented policy* would be

$$\hat{\pi}_k^p = \bar{X}\log\left(\frac{s}{c}\right), \quad k = 1,2,\ldots,m.$$

with profit

$$\hat{\psi}(Y) = \sum_{k=1}^{m}\left\{s\min\left\{Y_k, \bar{X}\log\left(\frac{s}{c}\right)\right\} - c\bar{X}\log\left(\frac{s}{c}\right)\right\},$$

where $\bar{X} = (1/n)\sum_{k=1}^{n} X_k$. If we use the updated estimate of θ at the beginning of period k, we get

$$\hat{\pi}_k^p = \bar{X}_k\log\left(\frac{s}{c}\right), \quad k = 1,2,\ldots,m.$$

With this implementation,

$$\hat{\psi}(Y) = \sum_{k=1}^{m}\left\{s\min\left\{Y_k, \bar{X}_k\log\left(\frac{s}{c}\right)\right\} - c\bar{X}_k\log\left(\frac{s}{c}\right)\right\},$$

and it can be easily verified that (see Liyanage and Shanthikumar [80])

$$\lim_{m\to\infty}\frac{1}{m}\hat{\psi}(Y) = s\theta\left(1-\left(\frac{n}{n+\log(s/c)}\right)^n\right) - c\theta\log\left(\frac{s}{c}\right).$$

Observe that the average profit achieved is smaller than the expected profit $(s-c)\theta - c\theta\ln(s/c)$. For small values of n, this loss can be substantial. For example, when $n=4$ and $s/c=1.2$, the percent loss over the optimal value with known θ is 22.86. (see Liyanage and Shanthikumar [80], p. 343). When the demand is nonstationary, we will be forced to use a moving average or exponential smoothing to forecast the future demand. In such a case, we will need to use a small value for n.

Subjective Bayesian Approach. Under the subjective Bayesian approach, given $\mathcal{I}_0$, one assumes that the parameter characterizing the measure is random and postulates a distribution for that parameter (Θ). Suppose we assume that the density function of Θ is $f_\Theta(\theta)$, $\theta\in\Theta$, and the conditional density of $\{\Theta \mid \mathbf{X}\}$ as $f_{\Theta\mid\mathbf{X}}(\theta\mid\mathbf{X})$, $\theta\in\Theta$. The objective function in this case is

$$E_\Theta[\phi(\pi,\Theta)\mid\mathbf{X}] = \int_{\theta\in\Theta}\phi(\pi,\theta)f_{\Theta\mid\mathbf{X}}(\theta\mid\mathbf{X})d\theta.$$

Let

$$\pi^B_{f_\Theta}(\mathbf{X}) = \arg\max\{E_\Theta[\phi(\pi,\Theta) \mid \mathbf{X}]\colon \pi \in \Gamma\}$$

and

$$\phi^B_{f_\Theta}(\theta) = E_{\mathbf{X}}[\phi(\pi^B_{f_\Theta}(\mathbf{X}),\theta)].$$

The Inventory Rat (cont'd.). Often, the subjective prior is chosen to be the conjugate of the demand distribution (e.g., Azoury [10]). When the demand is exponentially distributed, we should choose the Gamma prior for the unknown rate, say $\lambda = 1/\theta$ of the exponential distribution (e.g., Robert [82], p. 121). So, let (for $\alpha, \beta > 0$)

$$f_\Theta(\theta) = \frac{(\beta/\theta)^{\alpha+1}}{\beta\Gamma(\alpha)} \exp\left\{-\frac{\beta}{\theta}\right\}, \quad \theta \geq 0.$$

Note that $E[\Lambda] = E[1/\Theta] = \alpha/\beta$. We still need to choose the parameters α and β for this prior distribution. Straightforward algebra will reveal that

$$\pi^B_{f_\Theta}(\mathbf{X}) = (\beta + n\bar{X})\left(\left(\frac{s}{c}\right)^{1/(\alpha+n)} - 1\right).$$

Even if the demand distribution is exponential, if the demand mean is nonstationary, the Bayesian estimate will converge to an incorrect parameter value. Hence, we need to reinitiate the prior distribution every now and then. Suppose we do that every n periods. Then

$$\pi^B_{k:f_\Theta}(\mathbf{X}) = (\beta + n\bar{X}_k)\left(\left(\frac{s}{c}\right)^{1/(\alpha+n)} - 1\right), \quad k = 1, 2, \ldots, m,$$

with profit

$$\hat{\psi}(Y) = \sum_{k=1}^{m}\left\{s\min\left\{Y_k, (\beta + n\bar{X}_k)\left(\left(\frac{s}{c}\right)^{1/(\alpha+n)} - 1\right)\right\} - c(\beta + n\bar{X}_k)\left(\left(\frac{s}{c}\right)^{1/(\alpha+n)} - 1\right)\right\}.$$

With this implementation, it can be verified that

$$\lim_{m\to\infty}\frac{1}{m}\hat{\psi}(Y) = s\theta\left(1 - \left(\frac{\theta}{(s/c)^{1/(\alpha+n)} + \theta - 1}\right)^n \exp\left\{-\frac{\beta}{\theta}\left(\left(\frac{s}{c}\right)^{1/(\alpha+n)} - 1\right)\right\}\right) - c(\beta + n\theta)\left(\left(\frac{s}{c}\right)^{1/(\alpha+n)} - 1\right).$$

For bad choices of α and β, the performance can be poor. The success of this policy will depend on a lucky guess for α and β.

2.2.2. Nonparametric Modeling. Suppose we have characterized P without making any assumptions regarding the parametric form of $\mathbf{Y}$. Now define

$$\phi^g(\pi, P) = E[\psi(\pi, \mathbf{Y})],$$

and solve

$$\phi^g(P) = \max_{\pi\in\Gamma}\{\phi(\pi, P)\},$$

and choose a $\pi^g(P) \in \Gamma$ such that

$$\psi(\pi^g(P), P) = \phi^g(P).$$

The Inventory Rat (cont'd.). Observe that the optimal order quantity $\pi^g(F_Y)$ for demand distribution F_Y is given by

$$\pi^g(F_Y) = \bar{F}^{\text{inv}}_Y\left(\frac{c}{s}\right),$$

where $\bar{F}_Y^{\text{inv}}$ is the inverse of the survival function ($\bar{F}_Y = 1 - F_Y$) of the demand. We may, therefore, use the empirical demand distribution ($\hat{\bar{F}}_Y$) to obtain an estimate of the order quantity. Let $X_{[0]} = 0$ and $X_{[r]}$ be the r-th order statistic of $\{X_1, \ldots, X_n\}$, $r = 1, 2, \ldots, n$. Because the demand is assumed to be continuous, we set

$$\hat{\bar{F}}_Y(x) = 1 - \frac{1}{n}\left\{ r - 1 + \frac{x - X_{[r-1]}}{X_{[r]} - X_{[r-1]}} \right\}, \quad X_{[r-1]} < x \le X_{[r]}, \quad r = 1, 2, \ldots, n.$$

Then, the implemented order quantity $\hat{\pi}^g$ based on the empirical distribution is

$$\hat{\pi}^g = \hat{\bar{F}}_X^{\text{inv}}\left(\frac{c}{s}\right) = X_{[\hat{r}-1]} + \hat{a}(X_{[\hat{r}]} - X_{[\hat{r}-1]}),$$

where $\hat{r} \in \{1, 2, \ldots, n\}$ satisfies

$$n\left(1 - \frac{c}{s}\right) < \hat{r} \le n\left(1 - \frac{c}{s}\right) + 1,$$

and

$$\hat{a} = n\left(1 - \frac{c}{s}\right) + 1 - \hat{r}.$$

It can be shown that (see Liyanage and Shanthikumar [80], p. 345),

$$\lim_{m\to\infty} \frac{1}{m}\hat{\psi}(\mathbf{Y}) = c\theta\left\{ \frac{s}{c}\left(1 - \left(\frac{n - \hat{r} + 2}{n+1}\right)\left(\frac{n - \hat{r} + 1}{n - \hat{r} + 1 + \hat{a}}\right)\right) - \sum_{k=1}^{\hat{r}-1} \frac{1}{n - k + 1} - \frac{\hat{a}}{n - \hat{r} + 1} \right\}.$$

The loss in expected profit in this case can be substantially bad. For example, when $n = 4$ and $s/c = 1.2$, the percent loss over the optimal value with known θ is 73.06. (This is much worse than the 22.86 % loss with the use of the sample mean for this example.)

It is clear that with limited and/or nonstationarity in the underlying stochastic process, we may have significant errors in our models due to errors in the statistical assumptions we used for the parametric or nonparametric models and due to estimation errors. Therefore, we should find robust solutions to these errors. We could do this by attending to two issues: (1) find ways to incorporate these errors in the model itself, and (2) find a way to obtain a robust solution.

3. Model Uncertainty and Flexible Modeling

From the preceding discussion, it is clear that we have to account for the errors we will have in calibrating the stochastic model. Therefore, we will not know the exact probability measure for our model. Given this it is reasonable to argue that one should not make a decision based only on a single model (that is, using a single probability measure). Under flexible modeling, we would consider a collection of models and modify our assumption.

Modified Assumption 1 (A1). The chosen collection of probability measures $\mathcal{P}$ contains the true probability measure P_0 or one that is very close (in some sense) to it.

It is up to us now to define this collection of measures. Following tradition, we will have three different approaches one could take to develop models of model uncertainty.

3.1. Flexible Modeling with a Variable Uncertainty Set

If the goal is to keep the resulting optimization problem within a class that has efficient solution algorithms or strong approximations, one may consider a collection of degenerate probability measures. That is, one considers

$$\mathcal{P} = \{P_\omega^d, \omega \in \Omega\}.$$

This is essentially to identify the possible values that $\mathbf{Y}$ can take. Let $\mathcal{Y}$ be this state space. Then one considers a collection of problems

$$\psi(\pi, Y), \quad Y \in \mathcal{Y}.$$

It is easy to see that in almost all real problems, the probability measure P_0 will not be in $\mathcal{P}$. Yet, a vast majority of robust optimization reported in the OR/MS literature follows this modeling approach (e.g., Atamturk [5], Atamturk and Zhang [6], Averbakh [7, 8, 9], Ben-Tal and Nemirovski [11, 12, 13, 14], Bertsimas and Sim [20, 21, 22], Bertsimas and Thiele [23], Bertsimas et al. [24], Kouvelis and Yu [70], Soyster [86]).

3.2. Flexible Modeling with a Parametric Uncertainty Set

Suppose our statistical assumptions are valid, and the only unknown are the true parameter values. Then, the collection of measures we consider could be

$$\mathcal{P} = \{P_\theta^p, \theta \in \Theta\},$$

for some set Θ of parameter values. Then, one considers a collection of problems

$$\phi^p(\pi, \theta), \quad \theta \in \Theta.$$

This appears to be a very promising way to formulate and solve real problems. Application of this approach to portfolio optimization is discussed in Lim et al. [76, 78].

3.3. Flexible Modeling with a Nonparametric Uncertainty Set

For flexible modeling with a nonparametric uncertainty set, we first identify a nominal model (or probability measure, say, $\hat{P}$). Then the collection of models are chosen to be a closed ball around this nominal model. Let $d(P, \hat{P})$ be some distance measure between P and $\hat{P}$. If the measures are fully characterized by a density (or distribution) function, the distance will be defined with respect to the density (or distribution) functions. The collection of models thus considered will be

$$\mathcal{P} = \{P\colon d(P, \hat{P}) \le \alpha\},$$

where α is the minimum deviation that we believe is needed to assure that the true probability measure P_0 is in $\mathcal{P}$. Some distance measures commonly used are listed below.

3.3.1. Distance Measures for Density Functions. We will specify the different types of distances for the density functions of continuous random variables. Analogous distances can be defined for discrete random variables as well.

Kullback-Leibler Divergence (Relative Entropy)

$$d_{KL}(f, \hat{f}) = \int_x f(x) \log\left(\frac{f(x)}{\hat{f}(x)}\right) dx.$$

It is easy to verify that d_{KL} takes values in $[0, \infty]$ and is convex in f. However, it is not a metric (it is not symmetric in $(f, \hat{f})$ and does not satisfy the triangle inequality). One very useful property of d_{KL} is that it is sum separable for product measures. This comes in very handy in dynamic optimization with model uncertainty.

Hellinger Distance

$$d_H(f, \hat{f}) = \sqrt{\frac{1}{2}}\left[\int_x \left(\sqrt{f(x)} - \sqrt{\hat{f}(x)}\right)^2 dx\right]^{1/2}.$$

Hellinger distance as defined above is a metric that takes a value in $[0, 1]$. One useful property of this metric in dynamic optimization is that the Hellinger affinity $(1 - d_H^2)$ is product separable for product measures.

Chi-Squared Distance

$$d_{CS}(f,\hat{f}) = \int_x \frac{(f(x)-\hat{f}(x))^2}{\hat{f}(x)}\,dx.$$

Discrepancy Measure

$$d_D(f,\hat{f}) = \sup\left\{ \left| \int_a^b (f(x)-\hat{f}(x))\,dx \right| \colon a<b \right\}.$$

Total Variation Distance

$$d_{TV}(f,\hat{f}) = \frac{1}{2}\sup\left\{ \int_x h(x)(f(x)-\hat{f}(x))\,dx \colon |h(x)|\le 1 \right\}.$$

Wasserstein (Kantorovich) Metric

$$d_W(f,\hat{f}) = \sup\left\{ \int_x h(x)(f(x)-\hat{f}(x))\,dx \colon |h(x)-h(y)| \le |x-y| \right\}.$$

3.3.2. Distance Measures for Cumulative Distribution Functions.

Kolmogorov (Uniform) Metric

$$d_K(F,\hat{F}) = \sup\{|F(x)-\hat{F}(x)| \colon x\in\mathcal{R}\}.$$

Levy (Prokhorov) Metric

$$d_L(F,\hat{F}) = \inf\{h \colon F(x-h)-h \le \hat{F}(x) \le F(x+h)+h;\ h>0;\ x\in\mathcal{R}\}.$$

Wasserstein (Kantorovich) Metric

$$d_W(F,\hat{F}) = \int_x |F(x)-\hat{F}(x)|\,dx.$$

3.3.3. Distance Measures for Measures.

Kullback-Leibler Divergence (Relative Entropy)

$$d_{KL}(P,\hat{P}) = \int_\Omega \log\left(\frac{dP}{d\hat{P}}\right) dP.$$

Prokhorov Metric

Suppose Ω is a metric space with metric d. Let $\mathcal{B}$ be the set of all Borel sets of Ω, and for any $h>0$, define $B^h = \{x \colon \inf_{y\in B} d(x,y) \le h\}$ for any $B\in\mathcal{B}$. Then,

$$d_P(P,\hat{P}) = \inf\{h \mid P(B) \le P(B^h)+h;\ h>0;\ B\in\mathcal{B}\}.$$

Discrepancy Measure

Suppose Ω is a metric space with metric d. Let $\mathcal{B}^c$ be the collection of all closed balls in Ω.

$$d_D(P,\hat{P}) = \sup\{|P(B)-\hat{P}(B))| \colon B\in\mathcal{B}^c\}$$

Total Variation Distance

$$d_{TV}(P,\hat{P}) = \sup\{|P(A)-\hat{P}(A)| \colon A\subset\Omega\}.$$

Wasserstein (Kantorovich) Metric

Suppose Ω is a metric space with metric d.

$$d_W(P,\hat{P}) = \sup\left\{ \int_\Omega h(\omega)(P(d\omega)-\hat{P}(d\omega)) \colon |h(x)-h(y)| \le d(x,y), x,y\in\Omega \right\}$$

The majority of the flexible modeling in finance is done using uncertainty sets for measures (e.g., Hansen and Sargent [62] and its references). Application of this approach to dynamic programming is given in Iyengar [66] and in revenue management in Lim and Shanthikumar [73] and Lim et al. [77].

4. Robust Optimization

Now that we have a collection of models, we need to decide how to find a very good solution for the true model. For this, we assume that our robust optimization will give such a good solution.

Modified Assumption 2 (A2). The robust solution (optimal in some sense) obtained with the collection of measures $\mathcal{P}$ leads to a performance that is either optimal or close to optimal (in some sense) with respect to P_0.

4.1. Max-Min Objective

The most commonly used approach to finding a (so-called) robust solution for the given set of models is to find the best solution to the worst model among the collection of models. The optimization problem is

$$\phi^r = \max_{\pi\in\Gamma}\left\{\min_{P\in\mathcal{P}}\{\phi(\pi,P)\}\right\}.$$

And the solution sought is

$$\pi^r = \arg\max_{\pi\in\Gamma}\min_{P\in\mathcal{P}}\{\phi(\pi,P)\}.$$

If the true model is the worst one, then this solution will be satisfactory. However, if the true model is the best one or something close to it, this solution could be very bad (that is, the solution need not be robust to model error at all). As we will soon see, this can be the case. However, this form of (so-called) robust optimization is still very popular, because the resulting optimization tends to preserve the algorithmic complexity very close to that of the original single model case. However, if we really want a robust solution, its performance needs to be compared to what could have been the best for every model in the collection. This idea of benchmarking will be discussed later. Let us now look at the inventory example:

The Inventory Rat (cont'd.). We will now apply max-min robust optimization to the inventory rat with the three different flexible modeling ideas.

Uncertainty Set for Demand. Suppose the demand can take a value in $[a,b]$. That is, $a \le Y_k \le b$, $k = 1,2,\ldots,m$. Then we have the robust optimization problem

$$\phi^r = \max_{\pi_k\ge 0}\left\{\min_{a\le Y_k\le b}\sum_{k=1}^{m}\{s\min\{Y_k,\pi_k\} - c\pi_k\}\right\}.$$

Because the inner minimization is monotone in Y_k, it is immediate that

$$\phi^r = \max_{\pi_k\ge 0}\sum_{k=1}^{m}\{s\min\{a,\pi_k\} - c\pi_k\} = (s-c)ma,$$

and

$$\pi_k^r = a, \quad k = 1,2,\ldots,m.$$

Clearly, this a very pessimistic solution (for example, if $a = 0$). Specifically, if the true demand happens to be b, the performance of this solution will be the worst. Furthermore, observe that the solution is independent of s and c.

Uncertainty Set for the Mean of Exponentially Distributed Demand. Suppose the mean demand can take a value in $[a,b]$. That is, $a \le E[Y_k] = \theta \le b$, $k = 1,2,\ldots,m$. Then, we have the robust optimization problem

$$\phi^r = \max_{\pi_k\ge 0}\left\{\min_{a\le\theta\le b}\sum_{k=1}^{m}\{s\theta(1-\exp\{-\pi_k/\theta\}) - c\pi_k\}\right\}.$$

As before, the inner minimization is monotone in θ, and it is immediate that

$$\phi^r = \max_{\pi_k \geq 0} \sum_{k=1}^{m} \left\{ sa\left(1 - \exp\left\{-\frac{\pi_k}{a}\right\}\right) - c\pi_k \right\} = \left((s-c)a - ca\log\left(\frac{s}{c}\right)\right)m$$

and

$$\pi_k^r = a\log\left(\frac{s}{c}\right), \quad k = 1, 2, \ldots, m.$$

Clearly, this, too, is a very pessimistic solution (for example, if $a = 0$). If the true mean demand happens to be b, the performance of this solution will be the worst.

Uncertainty Set for Density Function of Demand. Suppose we choose the Kullback-Leibler Divergence (Relative Entropy) to define the collection of possible demand density functions. Suppose the nominal model chosen is an exponential distribution with mean $\hat{\theta}$. That is,

$$\hat{f}(x) = \frac{1}{\hat{\theta}} \exp\left\{-\frac{1}{\hat{\theta}}x\right\}, \quad x \geq 0.$$

Then, the collection of density functions for the demand is

$$\mathcal{P} = \left\{ f\colon \int_{x=0}^{\infty} f(x)\log\left(\frac{f(x)}{\hat{f}(x)}\right)dx \leq \alpha;\ \int_{x=0}^{\infty} f(x)\,dx = 1;\ f \geq 0 \right\}.$$

The min-max robust optimization is then

$$\max_{\pi \geq 0} \min_{f \in \mathcal{P}} \left\{ s\int_{x=0}^{\pi} \left\{ \int_{z=x}^{\infty} f(z)\,dz \right\} dx - c\pi \right\}.$$

Defining $\kappa(x) = f(x)/\hat{f}(x)$ and considering the Lagrangian relaxation of the above problem, one obtains (with $\beta \geq 0$),

$$\max_{\pi \geq 0} \text{-} \min_{\kappa \geq 0} \left\{ s\int_{x=0}^{\pi} \left\{ \int_{z=x}^{\infty} \kappa(x)\hat{f}(z)\,dz \right\} dx - c\pi \right.$$
$$\left. +\beta \int_{x=0}^{\infty} \kappa(x)\log(\kappa(x))\hat{f}(x)\,dx\colon \int_{x=0}^{\infty} \kappa(x)\hat{f}(x)\,dx = 1 \right\}.$$

It can be verified that the solution to the above relaxation is

$$\kappa(x) = \frac{(s-c)\hat{\theta} + \beta}{\beta} \exp\{-sx\}, \quad 0 \leq x \leq \pi^r,$$

$$\kappa(x) = \frac{(s-c)\hat{\theta} + \beta}{\beta} \exp\{-sy\}, \quad \pi^r \leq x,$$

and

$$\pi^r = \hat{\theta}\left\{\log\left(\frac{s}{c}\right) + \log\left(\frac{(s-c)\hat{\theta} + \beta}{\beta}\right)\right\}\left\{\frac{\beta}{\beta + s\hat{\theta}}\right\}.$$

Furthermore, it can be shown that the solution to the original problem is obtained by choosing β such that

$$\int_{x=0}^{\infty} \kappa(x)\log(\kappa(x))\hat{f}(x)\,dx = \alpha.$$

It can be shown that β monotonically decreases as a function of α with $\beta \to 0$ as $\alpha \to \infty$, and $\beta \to \infty$ as $\alpha \to 0$. Notice that the robust order quantity goes to zero as $\beta \to 0$ (that is, when $\alpha \to \infty$), and the order quantity becomes the nominal order quantity $\hat{\theta}\log(s/c)$ when $\beta \to \infty$ (that is, when $\alpha \to 0$). Clearly, in the former case, we allow a demand that is zero with probability one, and in the latter case, we restrict the collection of models to the nominal one.

All three formulations suffer because the inner minimization is monotone and the worst model is chosen to optimize. In what follows, we will see that the idea of using benchmarks will overcome this shortcoming.

4.2. Min-Max Regret Objectives, Utility, and Alternative Coupling with Benchmark

Recall that $\phi^g(P)$ is the optimal objective function value we can achieve if we knew the probability measure P. Hence, we may wish to find a solution that gives an objective function value that comes close to this for all measures in $\mathcal{P}$. Hence, we consider the optimization problem

$$\phi^r = \min_{\pi\in\Gamma}\Big\{\max_{P\in\mathcal{P}}\{\phi^g(P) - \phi(\pi, P)\}\Big\},$$

and the solution sought is

$$\pi^r = \arg\min_{\pi\in\Gamma}\max_{P\in\mathcal{P}}\{\phi^g(P) - \phi(\pi, P)\}.$$

One may also wish to see how the robust policy works with respect to the optimal policy with the actual profit and not its expectation. Given that one has a utility function U^r for this deviation, the coupled objective function is

$$\phi^r = \min_{\pi\in\Gamma}\Big\{\max_{P\in\mathcal{P}}\{E_P[U^r(\psi(\pi^g(P), \mathbf{Y}) - \psi(\pi, \mathbf{Y}))]\}\Big\},$$

and the solution sought is

$$\pi^r = \arg\min_{\pi\in\Gamma}\max_{P\in\mathcal{P}}\{E_P[U^r(\psi(\pi^g(P), \mathbf{Y}) - \psi(\pi, \mathbf{Y}))]\}.$$

The Inventory Rat (cont'd.). Observe that clairvoyant ordering will result in a profit of $(s-c)Y$. Hence, if we order π units, the regret is $(s-c)Y - \{s\min\{\pi, Y\} - c\pi\} = s\max\{Y-\pi, 0\} - c(Y-\pi)$. Hence, we wish to solve

$$\min\max_{a\le Y\le b}\{s\max\{Y-\pi, 0\} - c(Y-\pi)\}.$$

The optimal solution is

$$\pi^r = a + (b-1)\left(\frac{s-c}{s}\right).$$

Unlike in the min-max robust optimization, here, the order quantity depends on s and c.

4.3. Max-Min Competitive Ratio Objective with Alternative Coupling with Benchmark

Suppose $\phi^g(P) \ge 0$ for all $P \in \mathcal{P}$. Then, instead of looking at the difference in the objective function values, we may wish to look at the ratios (and find a solution that achieves a ratio close to one for all P). Hence, we consider the optimization problem

$$\phi^r = \min_{\pi\in\Gamma}\left\{\max_{P\in\mathcal{P}}\left\{\frac{\phi(\pi, P)}{\phi^g(P)}\right\}\right\},$$

and the solution sought is

$$\pi^r = \arg\min_{\pi\in\Gamma}\max_{P\in\mathcal{P}}\left\{\frac{\phi(\pi, P)}{\phi^g(P)}\right\}.$$

One may also wish to see how the robust policy works with respect to the optimal policy with the actual profit, and not its expectation. Suppose $\psi(\pi^g(P), \mathbf{Y}) \ge 0$. Given that one has a utility function U^r for this deviation, the coupled objective function is

$$\phi^r = \min_{\pi\in\Gamma}\left\{\max_{P\in\mathcal{P}}\left\{E_P\left[U^r\left(\frac{\psi(\pi, \mathbf{Y})}{\psi(\pi^g(P), \mathbf{Y})}\right)\right]\right\}\right\},$$

and the solution sought is

$$\pi^r = \arg\min_{\pi\in\Gamma}\max_{P\in\mathcal{P}}\left\{E_P\left[U^r\left(\frac{\psi(\pi, \mathbf{Y})}{\psi(\pi^g(P), \mathbf{Y})}\right)\right]\right\}.$$

5. Classical Statistics and Flexible Modeling

We will now discuss how classical statistics can be used to characterize model uncertainty of different types. To do this, first we have to postulate a statistical model for $\mathbf{X}, \mathbf{Y}$. Suppose the extended measure for this is P^e (note that, then $P = \{P^e \mid \mathcal{I}_0\}$).

5.1. Predictive Regions and Variable Uncertainty Set

Let $\mathcal{S}_Y$ be the state space of $\mathbf{Y}$. Now, choose a predictive region $\mathcal{Y}(\mathbf{X}) \subset \mathcal{S}_Y$ for $\mathbf{Y}$ such that

$$P^e\{\mathbf{Y} \in \mathcal{Y}(\mathbf{X})\} = 1 - \alpha,$$

for some appropriately chosen value of α $(0 < \alpha < 1)$. Then, we could choose

$$\mathcal{Y} = \{\mathcal{Y}(\mathbf{X}) \mid \mathcal{I}_0\}.$$

The Inventory Rat (cont'd.). Suppose $\{X_1, X_2, \ldots, X_n, Y\}$ are i.i.d. exponential random variables with mean θ. Let χ^2_k be a Chi-squared random variable with k degrees of freedom, and $F_{r,s}$ be an F-random variable with (r, s) degrees of freedom. Then,

$$\frac{2n}{\theta}\bar{X} =^d \chi^2_{2n},$$

and

$$\frac{2}{\theta}Y =^d \chi^2_2.$$

Therefore

$$\frac{Y}{\bar{X}} =^d F_{2,\,2n},$$

and

$$P\{f_{2,\,2n,\,1-\alpha/2}\bar{X} \le Y \le f_{2,\,2n,\,\alpha/2}\bar{X}\} = 1 - \alpha,$$

where

$$P\{f_{2,\,2n,\,\beta} \le F_{2,\,2n}\} = \beta, \quad \beta \ge 0.$$

A $(1-\alpha)100\%$ predictive interval for Y is $(f_{2,\,2n,\,1-\alpha/2}\bar{X}, f_{2,\,2n,\,\alpha,2}\bar{X})$. Hence, with a min-max objective, the robust solution is (see §4.1)

$$\pi^r = f_{2,\,2n,\,1-\alpha/2}\bar{X}.$$

Observe that this implementation is independent of s and c. Alternatively, one may use a one-sided predictive interval $(f_{2,\,2n,\,1-\alpha}\bar{X}, \infty)$. Then

$$\pi^r = f_{2,\,2n,\,1-\alpha}\bar{X}.$$

This too is independent of s and c. Therefore, there is no guarantee that this solution will be robust to model uncertainty. Suppose we choose an α such that

$$1 - \alpha = P\left\{\left(\left(\frac{s}{c}\right)^{1/(1+n)} - 1\right)n \le F_{2,\,2n}\right\}.$$

Then

$$\pi^r = \left(\left(\frac{s}{c}\right)^{1/(1+n)} - 1\right)n\bar{X}.$$

Later, in operational learning, we will find that this is indeed the optimal order quantity when θ is unknown. It is, thus, conceivable that a good policy could be obtained using a deterministic robust optimization provided we have stable demand and sufficient data to test various α. If that is the case, then retrospective optimization using the past data would have yielded a very good solution anyway. The issue in this method of using min-max robust optimization is that the solution can be sensitive to the choice α, and that a good value for it cannot be chosen a priori. Hence, we need a robust optimization technique that is robust with respect to the choice of α.

5.2. Confidence Regions and Parameter Uncertainty Set

Let $t(\mathbf{X})$ be an estimator of θ. Now, choose a region $\mathcal{T}(\theta)$ such that

$$P^e\{t(\mathbf{X}) \in \mathcal{T}(\theta)\} = 1 - \alpha,$$

for some appropriately chosen value of α $(0 < \alpha < 1)$. Now define

$$\Theta(\mathbf{X}) = \{\theta\colon t(\mathbf{X}) \in \mathcal{T}(\theta)\}.$$

Then we could choose

$$\Theta = \{\Theta(\mathbf{X})|\mathcal{I}_0\}.$$

The Inventory Rat (cont'd). Suppose $\{X_1, X_2, \ldots, X_n, Y\}$ are i.i.d. exponential random variables with mean θ. Observing that

$$\frac{2n}{\theta}\bar{X} =^d \chi^2_{2n},$$

it is immediate that

$$P\left\{\frac{2n\bar{X}}{\chi^2_{2n,\,\alpha/2}} \le \theta \le \frac{2n\bar{X}}{\chi^2_{2n,\,1-\alpha/2}}\right\} = 1 - \alpha,$$

where

$$P\{\chi^2_{2n,\,\beta} \le \chi^2_{2n}\} = \beta, \quad \beta \ge 0.$$

A $(1-\alpha)100\%$ confidence interval for θ is $2n\bar{X}/\chi^2_{2n,\,\alpha/2}$, $2n\bar{X}/\chi^2_{2n,\,1-\alpha/2})$. Hence, with a min-max objective, the robust solution is (see §4.1)

$$\pi^r = \frac{2n\bar{X}}{\chi^2_{2n,\,\alpha/2}}.$$

Observe that this implementation is independent of s and c. Alternatively, one may use a one-sided predictive interval $(2n\bar{X}/\chi^2_{2n,\,\alpha}, \infty)$. Then

$$\pi^r = \frac{2n\bar{X}}{\chi^2_{2n,\,\alpha}}.$$

This, too, is independent of s and c.

6. Learning

Outside of Bayesian learning, the two popular techniques used for learning in decision making are (i) reinforcement learning (e.g., Sutton and Barto [81]) and (ii) statistical learning (e.g., Vapnik [90]). Applying either approache to the inventory rat problem results in a solution that is the same as in the nonparametric model discussed in §2.2.2 (see Jain et al. [67]), which we already know can result in poor results. We will not discuss these two approaches here.

6.1. Max-Min, Duality, and Objective Bayesian Learning

In this section, we will pursue the max-min benchmarking approach discussed earlier as a learning tool. Specifically, we will consider the dual problem, which can then be seen as a form of the objective Bayesian approach (see Berger [15], Robert [82]).

In a dynamic optimization scenario, it is the recognition that the implemented policy $\hat{\pi}_k$ at time k is a function of the past data $\mathbf{X}$ that motivates the need to incorporate learning in the optimization itself. Hence, in integrated learning and optimization, the focus is

$$\max_{\pi} E^e_\theta[\phi(\pi(\mathbf{X}), \theta)],$$

where the expectation over $\mathbf{X}$ is taken with respect to the probability measure P^e_θ.

This is indeed the focus of decision theory (Wald [91]), where minimization of a loss function is the objective. Naturally, one could define $-\phi$ as the risk function and apply the existing decision theory approaches to solve the above problem. It has already been recognized in decision theory that without further characterization of π, one may not be able to solve the above problem (e.g., Berger [15], Robert [82]). Otherwise, one could conclude that $\pi^p(\theta)$ is the optimal solution. Hence, one abides by the notion of an *efficient* policy π defined below.

Definition 1. A policy π_0 is efficient if there does not exist a policy π such that

$$E^e_\theta[\phi(\pi(\mathbf{X}),\theta)] \geq E^e_\theta[\phi(\pi_0(\mathbf{X}),\theta)], \quad \forall\,\theta,$$

with strict inequality holding for some values of θ.

Observe that $\pi_0 = \pi^p(\theta_0)$ for almost any θ_0 will be an efficient solution. Indeed, it is well known that any Bayesian solution $\pi^B(f_\Theta)$, if unique, is an efficient solution. Thus, one may have an unlimited number of efficient policies, and the idea of an efficient solution does not provide an approach to identifying a suitable policy. While it is necessary for a solution to be efficient, it is not sufficient (unless it is optimal).

Definition 2. A policy π_0 is optimal, if

$$E^e_\theta[\phi(\pi_0(\mathbf{X}),\theta)] \geq E^e_\theta[\phi(\pi(\mathbf{X}),\theta)], \quad \forall\,\theta,$$

for all π.

It is very unlikely that such a solution can be obtained without further restriction on π for real stochastic optimization problems. Consequently, in decision theory, one follows one of the two approaches. One that is commonly used in the OR/MS literature is to assume a prior distribution for the unknown parameter(s) (see §2.2.1). This eliminates any model uncertainty. However, this leaves one to have to find this prior distribution during implementation. This task may not be well defined in practice (see Kass and Wasserman [69]). To overcome this, there has been considerable work done on developing noninformative priors (e.g., Kass and Wasserman [69]). The relationship of this approach to what we will do in the next two sections will be discussed later. The second approach in decision theory is min-maxity. In our setting, it is

$$\max_\pi \min_\theta \{E^e_\theta[\phi(\pi(\mathbf{X}),\theta)]\}.$$

Unfortunately, though, in almost all applications in OR/MS, $E^e_{\mathbf{X}}[\phi(\pi(\mathbf{X}),\theta)]$ will be monotone in θ. For example, in the inventory problem, the minimum will be attained at $\theta = 0$. In general, suppose the minimum occurs at $\theta = \theta_0$. In such a case, the optimal solution for the above formulation is $\pi^p(\theta_0)$. Hence, it is unlikely that a direct application of the min-max approach of decision theory to the objective function of interest in OR/MS will be appropriate. Therefore, we will apply this approach using objectives with benchmark (see §§4.2 and 4.3 and also Lim et al. [75]). In this section, we will consider the relative performance

$$\eta(\pi,\theta) = \frac{\phi(\pi(\mathbf{X}),\theta)}{\phi^p(\theta)}.$$

The optimization problem now is

$$\eta^r = \max_\pi \min_\theta \{E^e_\theta[\eta(\pi(\mathbf{X}),\theta)]\}.$$

The dual of this problem (modulo some technical conditions; see Lim et al. [75]) is

$$\min_{f_\Theta}\text{-}\max_\pi \{E^e_\Theta[\eta(\pi(\mathbf{X}),\Theta)]\},$$

where f_Θ is a prior on the random parameter Θ of $\mathbf{X}$. For each given prior distribution f_Θ, the policy π that maximizes the objective η is the Bayesian solution. Let $\pi^B_{f_\Theta}$ be the solution and $\eta^B(f_\Theta)$ be the objective function value. Two useful results that relate the primal and the dual problems are (e.g., Berger [15]):

Lemma 1. *If*

$$\eta^B(f_\Theta) = \min_\theta \frac{E^e_\theta[\phi(\pi^B_{f_\Theta}(\mathbf{X}), \theta)]}{\phi^p(\theta)},$$

then $\pi^B_{f_\Theta}$ is the max-min solution to the primal and dual problems.

Lemma 2. *If $f^{(l)}_\Theta$, $l = 1, 2, \ldots$, is a sequence of priors and $\pi^B_{f_\Theta}$ is such that*

$$\lim_{l\to\infty} \eta^B(f^{(l)}_\Theta) = \min_\theta \frac{E^e_\theta[\phi(\pi^B_{f_\Theta}(\mathbf{X}), \theta)]}{\phi^p(\theta)},$$

then $\pi^B_{f_\Theta}$ is the max-min solution to the primal problem.

Now, we add a bound that apart from characterizing the goodness of a chosen prior f_Θ or the corresponding policy $\pi^B_{f_\Theta}$, will aid an algorithm in finding the max-min solution.

Lemma 3. *For any prior f_Θ,*

$$\min_\theta \frac{E^e_\theta[\phi(\pi^B_{f_\Theta}(\mathbf{X}), \theta)]}{\phi^p(\theta)} \le \eta^r \le \frac{\int_\theta E^e_\theta[\phi(\pi^B_{f_\Theta}(\mathbf{X}), \theta)] f_\Theta(\theta)\, d\theta}{\int_\theta \phi^p(\theta) f_\Theta(\theta)\, d\theta}.$$

6.2. Operational Learning

This section is devoted to describing how learning could be achieved through operational statistics. Operational statistics is introduced in Liyanage and Shanthikumar [80] and further explored in Chu et al. [35, 36]. The formal definition of operational statistics is given in Chu et al. [37].

In operational learning, we seek to improve the current practice in the implementation of the policies derived assuming the knowledge of the parameters. In this regard, let $\pi^p(\theta)$ be the policy derived, assuming that the parameter(s) are known. To implement, in the traditional approach, we estimate θ by, say, $\hat{\Theta}(\mathbf{X})$ and implement the policy $\hat{\pi}^p = \pi^p(\hat{\Theta}(\mathbf{X}))$. The corresponding expected profit is

$$\hat{\phi}^p(\theta) = E^e_\theta[\phi(\pi^p(\hat{\Theta}(\mathbf{X})), \theta)],$$

where the expectation over $\mathbf{X}$ is taken with respect to P^e_θ. In operational learning, first we identify a class of functions $\mathcal{Y}$ and a corresponding class of functions $\mathcal{H}$ such that

$$\hat{\Theta} \in \mathcal{Y}$$

and

$$\pi^p \circ \hat{\Theta} \in \mathcal{H}.$$

The second step is to choose a representative parameter value, say, θ_0, and solve

$$\max_{\pi\in\mathcal{H}} E^e_{\theta_0}[\phi(\pi(\mathbf{X}), \theta_0)]$$

subject to

$$E^e_\theta[\phi(\pi(\mathbf{X}), \theta)] \ge \hat{\phi}^p(\theta), \quad \forall\, \theta.$$

First, note that because $\pi^p \circ \hat{\Theta} \in \mathcal{H}$, we are guaranteed that a solution exists for the above optimization problem. Second, note that the selection of θ_0 is not critical. For it may happen that the selection of $\mathcal{H}$ is such that the solution obtained is independent of θ_0 (as we will see

in the inventory examples). Alternatively, we may indeed use a prior f_Θ on θ and reformulate the problem as

$$\max_{\pi\in\mathcal{H}} \int_\theta E^e_\theta[\phi(\pi(\mathbf{X}),\theta)] f_\Theta(\theta)\, d\theta$$

subject to

$$E^e_\theta[\phi(\pi(\mathbf{X}),\theta)] \geq \hat{\phi}^p(\theta), \quad \forall\, \theta.$$

It is also conceivable that alternative forms of robust optimization may be defined.

The Inventory Rat (cont'd.). Recall that $\pi^p(\theta) = \theta \log(s/c)$ and $\hat{\Theta}(\mathbf{X}) = \bar{\mathbf{X}}$. So, we could choose $\mathcal{H}$ to be the class of order-one-homogenous functions. Note that

$$\mathcal{H}_1 = \{\pi\colon \mathcal{R}^n_+ \to \mathcal{R}_+; \pi(\alpha\mathbf{x}) = \alpha\pi(\mathbf{x});\ \alpha \geq 0;\ \mathbf{x}\in\mathcal{R}^n_+\}$$

is the class of nonnegative order-one-homogeneous functions. Furthermore, observe that ψ is a homogeneous-order-one function (that is, $\psi(\alpha x, \alpha Y) = \alpha\psi(x, Y)$). Let Z be an exponential r.v. with mean 1. Then, $Y =^d \theta Z$, and one finds that ϕ, too, is a homogeneous-order-one function (that is, $\phi(\alpha x, \alpha\theta) = \alpha\phi(x,\theta)$).

Now, suppose we restrict the class of operational statistics π to homogeneous-order-one functions. That is, for some chosen θ_0, we consider the optimization problem

$$\max_{\pi\in\mathcal{H}_1} \{E^e_{\theta_0}[\phi(\pi(\mathbf{X}),\theta_0)]\}$$

subject to

$$E^e_\theta[\phi(\pi(\mathbf{X}),\theta)] \geq \hat{\phi}^p(\theta), \quad \forall\, \theta.$$

Let $Z_1, Z_2, \ldots, Z_n$ be i.i.d. exponential r.v.s with mean 1 and $\mathbf{Z} = (Z_1, Z_2, \ldots, Z_m)$. Then

$$\mathbf{X} =^d \theta\mathbf{Z}.$$

Utilizing the property that ϕ, π, and $\hat{\phi}^p$ are all homogeneous-order-one functions, we get

$$E^e_\theta[\phi(\pi(\mathbf{X}),\theta)] = \theta E^e_{\mathbf{Z}}[\phi(\pi(\mathbf{Z}),1)],$$

and $\hat{\phi}^p(\theta) = \theta\hat{\phi}^p(1)$. Hence, we can drop the constraints and consider

$$\max_{\pi\in\mathcal{H}_1} \{E^e_{\mathbf{Z}}[\phi(\pi(\mathbf{Z}),1)]\}.$$

Let $\mathbf{V}$ (with $|\mathbf{V}| = \sum_{k=1}^m V_k = 1$), and the dependent random variable R be defined such that

$$f_{R\,|\,\mathbf{V}}(r|\mathbf{v}) = \frac{1}{r^{n+1}} \frac{1}{(n-1)!} \exp\left\{-\frac{1}{r}\right\}, \quad r \geq 0,$$

and

$$f_{\mathbf{v}}(\mathbf{v}) = (n-1)!, \quad |\mathbf{v}| = 1; \quad \mathbf{v}\in\mathcal{R}^n_+.$$

Then

$$\mathbf{Z} =^d \frac{1}{R}\mathbf{V}.$$

Therefore

$$E_{\mathbf{Z}}[\phi(\pi(\mathbf{Z}),1)] = E_{\mathbf{V}}\left[E_R\left[\phi\left(\pi\left(\frac{\mathbf{V}}{R}\right),1\right)\,\middle|\,\mathbf{V}\right]\right].$$

Because we assumed π to be a homogeneous-order-one function, we get

$$E_{\mathbf{V}}\left[E_R\left[\phi\left(\pi\left(\frac{\mathbf{Z}}{R}\right),1\right)\,\middle|\,\mathbf{V}\right]\right] = E_{\mathbf{V}}\left[E_R\left[\frac{1}{R}\phi(\pi(\mathbf{V}),R)\right]\,\middle|\,\mathbf{V}\right]\right].$$

Hence, all we need to find the optimal operational statistics is to find

$$\pi^{os}(\mathbf{v}) = \arg\max\left\{ E_R\left[\frac{1}{R}\phi(\pi, R)\,\middle|\, \mathbf{V} = \mathbf{v}\right] : \pi \geq 0\right\}, \quad \mathbf{v} \in \mathcal{R}_+^n;\ |\mathbf{v}| = 1.$$

Then, the optimal homogenous-order-one operational statistic is (with $|\mathbf{x}| = \sum_{k=1}^n x_k$),

$$\pi^{os}(\mathbf{x}) = |\mathbf{x}|\pi^{os}\left(\frac{\mathbf{x}}{|\mathbf{x}|}\right), \quad \mathbf{x} \in \mathcal{R}_+^n.$$

After some algebra, one finds that (see Liyanage and Shanthikumar [80], Chu et al. [35]):

$$\pi^{os}(\mathbf{x}) = \left(\left(\frac{s}{c}\right)^{1/(1+n)} - 1\right)\sum_{k=1}^n x_k,$$

and

$$\hat{\phi}^{os}(\theta) = E_\theta[\phi(\pi^{os}(\mathbf{X}), \theta)] = \theta\left[c\left\{\frac{s}{c} - 1 - (n+1)\left(\left(\frac{s}{c}\right)^{1/(1+n)} - 1\right)\right\}\right].$$

This policy, compared to the classical approach, improves the expected profit by 4.96% for $n = 4$ and $s/c = 1.2$ (see Liyanage and Shanthikumar [80], p. 344).

7. Examples

7.1. Inventory Control with Observable Demand

Consider an inventory-control problem with instantaneous replenishment, backlogging, and finite planning horizon. Define the following input variables.

- m—number of periods in the planning horizon
- c—purchase price per unit
- s—selling price per unit
- $\{Y_1, Y_2, \ldots, Y_m\}$—demand for the next m periods
- b—backlogging cost per unit per period
- h—inventory carrying cost per unit per period

At the end of period m, all remaining inventory (if any) is salvaged (at a salvage value of c per unit). If at the end of period m orders are backlogged, then all orders are met at the beginning of period $m+1$. Let π_k ($\pi_k \geq 0$) be the order quantity at the beginning of period k ($k = 1, 2, \ldots, m$). Then, the total profit for the m periods is

$$\begin{aligned}\psi(\pi, \mathbf{Y}) = \sum_{k=1}^m \{-c\pi_k + s\{\max\{-W_{k-1}, 0\} + Y_k - \max\{-W_k, 0\}\}\} + c\max\{W_m, 0\} \\ + (s-c)\max\{-W_m, 0\} - \sum_{k=1}^m \{h\max\{W_k, 0\} + b\max\{-W_k, 0\}\},\end{aligned}$$

where $W_0 = 0$ and

$$W_k = W_{k-1} + \pi_k - Y_k, \quad k = 1, 2, \ldots, m.$$

Simple algebra reveals that

$$\psi(\pi, \mathbf{Y}) = \sum_{k=1}^m \psi_k(\pi_k, Y_k),$$

where

$$\psi_k(\pi_k, Y_k) = (s - c - b)Y_k + (b + h)\min\{W_{k-1} + \pi_k, Y_k\} - h(W_{k-1} + \pi_k), \quad k = 1, 2, \ldots, m.$$

Given $\mathcal{I}_k = \mathcal{F}_k$, we wish to find the optimal order quantity π_k^* for period k ($k = 1, \ldots, m$).

First, let us see what we can do if we are clairvoyant. Here, we will assume that all the future demand is known. It is not hard to see that

$$\pi_k^d(\omega_0) = Y_k(\omega_0), \quad k = 1, 2, \ldots, m,$$

and

$$\phi^d(\omega_0) = (s-c)\sum_{k=1}^{m} Y_k(\omega_0).$$

If we can implement this, then the profit experienced is $\hat{\psi}(\mathbf{Y}) = (s-c)\sum_{k=1}^{m} Y_k$ and the expected profit is $E[\hat{\psi}(\mathbf{Y})] = (s-c)m\theta$.

Suppose we assume that the future demand $\{Y_1, Y_2, \ldots, Y_m\}$ for the next m periods given $\mathcal{I}_0$ are i.i.d. with exponential density function with mean θ (that is, $f_Y(y) = (1/\theta)\exp\{-(1/\theta)y\}, y \geq 0$). Let

$$\phi_k(q,\theta) = E[(b+h)\min\{q, Y_k\} - hq] = (b+h)\theta\left(1 - \exp\left\{-\frac{q}{\theta}\right\}\right) - hq, \quad k = 1, 2, \ldots, m.$$

Then

$$q^*(\theta) = \arg\max\{\phi_k(q,\theta)\} = \theta\log\left(\frac{b+h}{h}\right).$$

It is then clear that

$$\pi_k(\theta) = q^*(\theta) - W_{k-1}, \quad k = 1, 2, \ldots, m,$$

and

$$\phi(\theta) = (s-c)m\theta - hm\theta\log\left(\frac{b+h}{h}\right).$$

If we use $\bar{X}$ as an estimate for the θ for implementing this policy, we get

$$\hat{\psi}(\mathbf{Y}) = (s-c-b)\sum_{k=1}^{m} Y_k + (b+h)\sum_{k=1}^{m}\min\left\{\bar{X}\log\left(\frac{b+h}{h}\right), Y_k\right\} - h\sum_{k=1}^{m}\bar{X}\log\left(\frac{b+h}{h}\right),$$

and an a priori expected profit of

$$\begin{aligned} E^e\left[\frac{1}{m}\hat{\psi}(\mathbf{Y})\right] = {} & (s-c)\theta - b\theta\left(\frac{n}{n+\log((b+h)/h)}\right)^n \\ & - h\theta\left(\left(\frac{n}{n+\log((b+h)/h)}\right)^n + \log\left(\frac{b+h}{h}\right) - 1\right). \end{aligned}$$

However, if we continue to update the estimate, we have

$$\hat{\pi}_k = \max\left\{\bar{X}_k\log\left(\frac{b+h}{h}\right) - W_{k-1}, 0\right\}, \quad k = 1, 2, \ldots, m,$$

and

$$\lim_{m\to\infty}\hat{\psi}(\mathbf{Y}) = E^e\left[\frac{1}{m}\hat{\psi}(\mathbf{Y})\right].$$

We will now apply operational learning to this problem (for details of this analysis, see Lim et al. [74]). Specifically, let $\mathcal{H}^1$ be the collection of order-one-homogeneous functions. Then, in operational learning, we are interested in

$$\max_{\pi_k\in\mathcal{H}^1}\sum_{k=1}^{m} E_\theta^e[\phi_k(\pi_k,\theta)],$$

where

$$\phi_k(\pi_k,\theta) = (b+h)E[\min\{W_{k-1}+\pi_k, Y_k\}] - hE[(W_{k-1}+\pi_k)],$$

$W_0 = 0$ and

$$W_k = W_{k-1} + \pi_k - Y_k, \quad k = 1, 2, \ldots, m.$$

First, we will consider the last period. Let $\mathbf{Y}^1$ be an empty vector and

$$\mathbf{Y}^k = (Y_1, \ldots, Y_{k-1}), \quad k = 2, \ldots, m.$$

Define the random vector $\mathbf{V}_m$ ($|\mathbf{V}_m| = 1$) and the dependent random variable R_m such that (see §6.2)

$$\frac{\mathbf{V}_m}{R_m} =^d (\mathbf{X}, \mathbf{Y}^m).$$

Now let

$$\tilde{\pi}_m(\mathbf{z}) = \arg\max\left\{ E_{R_m}\left[\frac{\phi_m(q, R_m)}{R_m} \,\middle|\, V_m = \mathbf{z} \right] : q \geq 0 \right\}, \quad \mathbf{z} \in \mathcal{R}_+^{n+m-1}, \quad |\mathbf{z}| = 1,$$

and

$$\tilde{\pi}_m(\mathbf{x}) = |\mathbf{x}| \tilde{y}_m\left(\frac{\mathbf{x}}{|\mathbf{x}|} \right), \quad \mathbf{x} \in \mathcal{R}_+^{n+m-1}.$$

Define

$$\pi_m(\mathbf{X}, \mathbf{Y}^m, w) = \max\{\tilde{y}_m(\mathbf{X}, \mathbf{Y}_m), w - Y_{m-1}\},$$

and

$$\phi_{m-1}^*(\mathbf{x}, q, \theta) = \phi_{m-1}(q, \theta) + E_{Y_{m-1}}[\phi_m(\pi_m(\mathbf{x}, Y_{m-1}, q), \theta)], \quad \mathbf{x} \in \mathcal{R}_+^{n+m-2}.$$

Having defined this for the last period, we can now set up the recursion for any period as follows: Define the random vector $\mathbf{V}_k$ ($|\mathbf{V}_k| = 1$) and the dependent random variable R_k such that

$$\frac{\mathbf{V}_k}{R_k} =^d (\mathbf{X}, \mathbf{Y}^k), \quad k = 1, 2, \ldots, m-1.$$

Now let

$$\tilde{\pi}_k(\mathbf{z}) = \arg\max\left\{ E_{R_k}\left[\frac{\phi_k^*(\mathbf{z}, q, R_k)}{R_k} \,\middle|\, V_k = \mathbf{z} \right] : q \geq 0 \right\}, \quad \mathbf{z} \in \mathcal{R}_+^{n+k-1}, \quad |\mathbf{z}| = 1,$$

and

$$\tilde{\pi}_k(\mathbf{x}) = |\mathbf{x}| \tilde{y}_m\left(\frac{\mathbf{x}}{|\mathbf{x}|} \right), \quad \mathbf{x} \in \mathcal{R}_+^{n+k-1}.$$

Define

$$\pi_k(\mathbf{X}, \mathbf{Y}^k, w) = \max\{\tilde{\pi}_k(\mathbf{X}, \mathbf{Y}^k), w - Y_{k-1}\},$$

and

$$\phi_{k-1}^*(\mathbf{x}, q, \theta) = \phi_{k-1}(q, \theta) + E_{Y_{k-1}}[\phi_k^*(y_k(\mathbf{x}, Y_{k-1}, q), 1)], \quad \mathbf{x} \in \mathcal{R}_+^{n+k-2}.$$

Now, the target inventory levels $\tilde{\pi}_k$ and the cost-to-go functions ϕ_{k-1}^* can be recursively computed starting with $k = m$. Computation of this operational statistics using numerical algorithms and/or simulation is discussed in Lim et al. [74].

7.2. Inventory Control with Sales Data

Let m, c, s, and $\{Y_1, Y_2, \ldots, Y_m\}$ be as defined earlier. At the end of each period, all remaining inventory (if any) is discarded (and there is no salvage value). Furthermore, any excess demand is lost, and lost demand cannot be observed. Let π_k ($\pi_k \geq 0$) be the order quantity at the beginning of period k ($k = 1, 2, \ldots, m$). Then, the total profit for the m periods is

$$\psi(\pi, \mathbf{Y}) = \sum_{k=1}^{m} \psi_k(\pi_k, Y_k),$$

where

$$\psi_k(\pi_k, Y_k) = sS_k - c\pi_k,$$

where $S_k = \min\{\pi_k, Y_k\}$ is the sales in period k, $k = 1, 2, \ldots, m$. Here, $\mathcal{I}_k(\pi) = \sigma(\{(S_j, \pi_j),\, j = 1, 2, \ldots, k\} \cup \mathcal{I}_0)$. We wish to find the optimal order quantity π_k^* for period k ($k = 1, \ldots, m$).

Suppose we assume that the future demand $\{Y_1, Y_2, \ldots, Y_m\}$ for the next m periods given $\mathcal{I}_0$ are i.i.d. with an exponential density function with mean θ (that is $f_Y(y) = (1/\theta)\exp\{-(1/\theta)y\}, y \geq 0$). If we know θ, this would then be exactly the same as the inventory rat problem. However, if θ is unknown (which will be the case in practise), we need to estimate it using possibly censored data. Suppose we have past demands, say, $\{X_1, \ldots, X_m\}$ and past sales $\{R_1, \ldots, R_m\}$. Let $I_k = I\{X_k = R_k\}$ be the indicator that the sales is the same as the demand in period k (which will be the case if we had more on-hand inventory than the demand). Given $(\mathbf{R}, \mathbf{I})$, the maximum likelihood estimator Θ_{MLE} of θ is (assuming that $\sum_{k=1}^{n} I_k \geq 1$, that is, at least once we got to observe the true demand)

$$\Theta_{MLE} = \frac{1}{\sum_{k=1}^{n} I_k} \sum_{k=1}^{n} R_k.$$

The implemented order quantities are then (assuming no further updates of the estimator)

$$\hat{\pi}_k = \Theta_{MLE} \log\left(\frac{s}{c}\right), \quad k = 1, 2, \ldots, m,$$

and the profit is

$$\hat{\psi}(\mathbf{Y}) = \sum_{k=1}^{m} \{s \min\{\Theta_{MLE} \log(s/c), Y_k\} - c\Theta_{MLE} \log(s/c)\}.$$

We will now show how operational learning can be implemented for a one-period problem ($m = 1$). Integrated learning for the multiperiod case can be done similar to the first example (see Lim et al. [74]). Suppose we are interested in

$$\max_{\pi \in \mathcal{H}^t} E^e_{\mathbf{X}}\{sE^e_{Y_1}[\min\{\pi, Y_1\}] - s\pi\},$$

for some suitably chosen class $\mathcal{H}^t$ of operational functions that includes the MLE estimator. This function also should allow us to find the solution without the knowledge of θ (what to do in operational learning if this is not possible is discussed in Chu et al. [37]). Because $R_k \leq X_k$ and $R_k = X_k$ when $I_k = 1$, and choosing a value of $X_k > R_k$ for $I_k = 0$, we could rewrite the MLE estimator as

$$\Theta_{MLE} = \frac{1}{\sum_{k=1}^{n} I\{X_k \leq R_k\}} \sum_{k=1}^{n} \min\{X_k, R_k\}.$$

Suppose $\mathcal{H}^t$ satisfies the following

$$\mathcal{H}^t = \{\eta\colon \mathcal{R}_+^n \times \mathcal{R}_+^n \Rightarrow \mathcal{R}_+;\ \eta(\alpha\mathbf{x}, \alpha\mathbf{r}) = \alpha\eta(\mathbf{x}, \mathbf{r}),\ \alpha \geq 0;\ \eta(\mathbf{y}, \mathbf{r}) = \eta(\mathbf{x}, \mathbf{r}),$$
$$\mathbf{y} = \mathbf{x} + (\alpha_1 I\{x_1 \geq r_1\}, \ldots, \alpha_n I\{x_n \geq r_n\}),\ \alpha \geq 0\}.$$

It is now easy to see that the function

$$h(\mathbf{x}, \mathbf{r}) = \frac{1}{\sum_{k=1}^n I\{x_k \leq r_k\}} \sum_{k=1}^n \min\{x_k, r_k\}$$

is an element of $\mathcal{H}^t$. Within this class of functions, the optimal operational statistics is

$$\pi(\mathbf{x}, \mathbf{r}) = \left(\left(\frac{s}{c}\right)^{1/(1+\sum_{k=1}^n I\{x_k \leq r_k\})} - 1\right) \sum_{k=1}^n \min\{x_k, r_k\}.$$

Hence, the operational order quantity is

$$\hat{\pi} = \left(\left(\frac{s}{c}\right)^{1/(1+\sum_{k=1}^n I_k)} - 1\right) \sum_{k=1}^n R_k.$$

Observe that if $I_k = 1$, $k = 1, 2, \ldots, n$ (that is, if there is no censoring), the above policy is identical to the policy for the newsvendor problem (see §6.2).

7.3. Portfolio Selection with Discrete Decision Epochs

We wish to invest in one or more of l stocks with random returns and a bank account with a known interest rate. Suppose at the beginning of period k, we have a total wealth of V_{k-1}. If we invest $\pi_k(i)V_{k-1}$ in stock i $(i = 1, 2, \ldots, l)$ and leave $(1 - \pi_k'\mathbf{e})V_{k-1}$ in the bank during period k, we will have a total wealth of

$$V_k(\pi_k) = Y_k(\pi_k)V_{k-1}$$

at the end of period k, $k = 1, 2, \ldots, m$. Here, $\pi_k = (\pi_k(1), \pi_k(2), \ldots, \pi_k(l))'$ and $\mathbf{e} = (1, 1, \ldots, 1)'$ is an l-vector of ones, and $Y_k(\pi_k) - 1$ is the rate of return for period k with a portfolio allocation π_k. The utility of the final wealth W_m for a portfolio selection π and utility function U is then

$$\psi(\pi, \mathbf{Y}) = U\left(v_0 \prod_{k=1}^m Y_k(\pi_k)\right).$$

where v_0 initial wealth at time 0.

We will now discuss how we traditionally complete these models, find the optimal policies, and *implement* them. Naturally, to complete the modeling, we need to define a probability measure P on $(\Omega, \mathcal{F}, (\mathcal{F}_k)_{k\in\mathcal{M}})$ given $\mathcal{I}_0$ and decide the sense (usually in the sense of expectation under P) in which the reward function is maximized. In these examples, almost always we simplify our analysis further by assuming a parametric family for F_Y.

We will first describe the classical continuous time model, which we will use to create our discrete time parametric model $Y_k(\pi_k)$, $k = 1, 2, \ldots, m$. Suppose the price process of stock i is $\{S_t(i), 0 \leq t \leq m\}$ given by

$$dS_t(i) = (\mu_t(i) + \sigma_t'(i)dW_t)S_t(i), \quad 0 \leq t \leq m, \quad i = 1, 2, \ldots, l,$$

where $\{W_t, 0 \leq t \leq m\}$ is a vector-valued diffusion process, $\mu_t(i)$ is the drift, and $\sigma_t(i)$ are the volatility parameters of stock i, $i = 1, 2, \ldots, l$. Let r_t, $0 \leq t \leq m$ be the known interest rate. Suppose the value of the portfolio is $V_t(\pi)$ at time t under a portfolio allocation policy π.

Under π, the value of investments in stock i at time t is $\pi_t(i)V_t(\pi)$. The money in the bank at time t is $(1-\pi_t\mathbf{e})V_t(\pi)$. Then, the wealth process $V_t(\pi)$ evolves according to

$$dV_t(\pi) = V_t(\pi)\{(r_t + \pi_t'\mathbf{b}_t)dt + \pi_t'\sigma_t' dW_t\}, \quad 0 \le t \le m,$$

where $b_t(i) = \mu_t(i) - r_t$, $i = 1, 2, \ldots, l$ and $V_0(\pi) = v_0$.

Now, suppose we are only allowed to decide on the ratio of portfolio allocation at time $k-1$, and the same ratio of allocation will be maintained during $[k-1, k)$, $k = 1, 2, \ldots, m$. In the classical continuous time model, now assume that $\mu_t = \mu_k$; $\sigma_t = \sigma_k$ and $\pi_t = \pi_k$, $k-1 \le t < k$, $k = 1, 2, \ldots, m$. Then, the utility at $T = m$ is

$$\psi(\pi, \mathbf{Z}) = U\left(v_0 \prod_{k=1}^{m} \exp\left\{r_k + \pi_k'\mathbf{b}_k - \frac{1}{2}\pi_k Q_k \pi_k + \pi_k'\sigma_k Z_k\right\}\right),$$

where $Q_k = \sigma_k\sigma_k'$ and $\{Z_k, k = 1, 2, \ldots, m\}$ are i.i.d. unit normal random vectors. Observe that the probability measure for this model is completely characterized by the parameters $(\mathbf{b}_k, \sigma_k)$, $k = 1, 2, \ldots, m$. We will assume that these parameters are independent of $\{Z_k, k = 1, 2, \ldots, m\}$ (though this assumption is not needed, we use them to simplify our illustration).

Suppose the values of parameters $(\mathbf{b}_k, \sigma_k)$, $k = 1, 2, \ldots, m$ are unknown, but we know a parameter uncertainty set for them. That is, $(\mathbf{b}_k, \sigma_k) \in \mathcal{H}_k$, $k = 1, 2, \ldots, m$. We wish to find a robust portfolio. We will use the robust optimization approach with competitive ratio objective with benchmarking. Specifically, we will now carry out the benchmarking with a log utility function. In this case, the benchmark portfolio is the solution of

$$\max_{\pi} E\log\left(v_0 \prod_{k=1}^{m} \exp\left\{r_k + \pi_k'\mathbf{b}_k - \frac{1}{2}\pi_k Q_k \pi_k + \pi_k'\sigma_k Z_k\right\}\right) \equiv \max_{\pi} \sum_{k=1}^{m} \left\{r_k + \pi_k'\mathbf{b}_k - \frac{1}{2}\pi_k' Q_k \pi_k\right\}.$$

It is not hard to see that

$$\pi_k^p = Q_k^{-1}\mathbf{b}_k, \quad k = 1, 2, \ldots, m,$$

and

$$V_m^p = v_0 \prod_{k=1}^{m} \exp\left\{r_k + \frac{1}{2}\mathbf{b}_k' Q_k^{-1}\mathbf{b}_k + \mathbf{b}_k' Q_k^{-1}\sigma_k Z_k\right\}.$$

Taking the ratio of V_m under a policy π and the benchmark value V_m^p, we find that the benchmarked objective is

$$\max_{\pi} \min_{(\mathbf{b}, \sigma) \in \mathcal{H}} E\left[U\left(\prod_{k=1}^{m} \frac{\exp\{r_k + \pi_k'\mathbf{b}_k' - \frac{1}{2}\pi_k' Q_k \pi_k + \pi_k'\sigma_k Z_k\}}{\exp\{r_k + \frac{1}{2}\mathbf{b}_k' Q_k^{-1}\mathbf{b}_k + \mathbf{b}_k' Q_k^{-1}\sigma_k Z_k\}}\right)\right].$$

This simplifies as

$$\max_{\pi} \min_{(\mathbf{b}, \sigma) \in \mathcal{H}} E\left[U\left(\prod_{k=1}^{m} \exp\left\{-\frac{1}{2}(\pi_k' - \mathbf{b}_k' Q_k^{-1}) Q_k (\pi_k - Q_k^{-1}\mathbf{b}_k) + (\pi_k' - \mathbf{b}_k' Q_k^{-1})\sigma_k Z_k\right\}\right)\right].$$

Observe that

$$E\left[\prod_{k=1}^{m} \exp\left\{-\frac{1}{2}(\pi_k' - \mathbf{b}_k' Q_k^{-1}) Q_k (\pi_k - Q_k^{-1}\mathbf{b}_k) + (\pi_k' - \mathbf{b}_k' Q_k^{-1})\sigma_k Z_k\right\}\right] = 1.$$

Furthermore,

$$\prod_{k=1}^{m} \exp\left\{-\frac{1}{2}(\pi_k' - \mathbf{b}_k' Q_k^{-1}) Q_k (\pi_k - Q_k^{-1}\mathbf{b}_k) + (\pi_k' - \mathbf{b}_k' Q_k^{-1})\sigma_k Z_k\right\}$$

is a log concave stochastic function. Hence, for any concave utility function U, the above objective can be rewritten as

$$\min_{\pi} \max_{(\mathbf{b},\sigma)\in\mathcal{H}} \sum_{k=1}^{m} (\pi_k' - \mathbf{b}_k' Q_k^{-1}) Q_k (\pi_k - Q_k^{-1}\mathbf{b}_k).$$

It now breaks into a sequence of single-period problems:

$$\sum_{k=1}^{m} \Big\{ \min_{\pi_k} \max_{(\mathbf{b}_k,\sigma_k)\in\mathcal{H}_k} (\pi_k' - \mathbf{b}_k' Q_k^{-1}) Q_k (\pi_k - Q_k^{-1}\mathbf{b}_k) \Big\}.$$

Given the uncertainty set $\mathcal{H}_k$, $k = 1, 2, \ldots, m$ the above robust optimization problem can be solved using duality (see Lim et al. [74]).

8. Summary and Conclusion

The interest in model uncertainty, robust optimization, and learning in the OR/MS areas is growing rapidly. The type of model uncertainties considered in the literature can be broadly categorized into three classes: Models with uncertainty sets for (1) variables, (2) parameters, and (3) measures. The robust optimization approaches used to find (robust or lack thereof) solutions falls into (a) min-max and (b) min-max with benchmarking. Two common ways to benchmark are through (1) regret and (2) competitive ratio. The main focus in OR/MS has been in the development of models with uncertainty sets for variables (deterministic models of model uncertainty) and deterministic min-max and min-max-regret robust optimization. Within this framework, the focus has been on developing efficient solution procedures for robust optimization. Only a very limited amount of work has been done on looking at stochastic models of model uncertainty and robust optimization with benchmarking. Very little is done in learning. We believe that a substantial amount of work needs to be done in the latter three topics.

Acknowledgments

This work was supported in part by the NSF Grant DMI-0500503 (for Lim and Shanthikumar) and by the NSF CAREER Awards DMI-0348209 (for Shen) and DMI-0348746 (for Lim).

References

[1] V. Agrawal and S. Seshadri. Impact of uncertainty and risk aversion on price and order quantity in the newsvendor problem. *Manufacturing and Service Operations Management* 2:410–423, 2000.

[2] S. Ahmed, U. Cakmak, and A. Shapiro. Coherent risk measures in inventory problems. Technical report, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 2005.

[3] E. W. Anderson, L. P. Hansen, and T. J. Sargent. Robustness, detection, and the price of risk. Technical report, New York University, New York, 2000.

[4] L. W. Anderson, P. Hansen, and T. J. Sargent A quartet of semigroups for model specification, robustness, price of risk, and model detection. *Journal of the European Economic Association* 1:68–123, 2003.

[5] A. Atamturk, Strong formulations of robust mixed 0-1 programming. *Mathematical Programming.* Forthcoming. 2006.

[6] A. Atamturk and M. Zhang. Two-stage robust network flow and design under demand uncertainty. *Operation Research.* Forthcoming. 2006.

[7] I. Averbakh. Minmax regret solutions for minmax optimization problems with uncertainty. *Operations Research Letters* 27:57–65, 2000.

[8] I. Averbakh. On the complexity of a class of combinatorial optimization problems with uncertainty. *Mathematical Programming* 90:263–272, 2001.

[9] I. Averbakh. Minmax regret linear resource allocation problems. *Operations Research Letters* 32:174–180, 2004.

[10] K. S. Azoury. Bayes solution to dynamic inventory models under unknown demand distribution. *Management Science* 31:1150–1160, 1985.

[11] A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research* 23:769–805, 1998.

[12] A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Operations Research Letters* 25:1–13, 1999.

[13] A. Ben-Tal and A. Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming A* 88:411–424, 2000.

[14] A. Ben-Tal and A. Nemirovski. Robust optimization—Methodology and applications. *Mathematical Programming B* 92:453–480, 2002.

[15] J. O. Berger. *Statistical Decision Theory and Bayesian Analysis*, 2nd ed. Springer, New York, 1985.

[16] P. Bernhard. A robust control approach to option pricing. M. Salmon, ed. *Applications of Robust Decision Theory and Ambiguity in Finance.* City University Press, London, UK, 2003.

[17] P. Bernhard. A robust control approach to option pricing, including transaction costs. A. S. Nowak and K. Szajowski, eds. *Advances in Dynamic Games, Annals of the International Society of Dynamic Games*, Vol 7. Birkhauser, 391–416, 2005.

[18] D. Bertsekas. *Convex Analysis and Optimization.* Athena Scientific, 2003.

[19] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming B* 98:49–71, 2003.

[20] D. Bertsimas and M. Sim. The price of robustness. *Operations Research* 52:35–53, 2004.

[21] D. Bertsimas and M. Sim. Robust discrete optimization under ellipsoidal uncertainty sets. Working paper, MIT, Cambridge, MA, 2004.

[22] D. Bertsimas and M. Sim. Tractable approximation to robust conic optimization problems. *Mathematical Programming* 107:5–36, 2006.

[23] D. Bertsimas and A. Thiele. A robust optimization approach to inventory theory. *Operations Research* 54:150–168, 2003.

[24] D. Bertsimas, D. Pachamanova, and M. Sim. Robust linear optimization under general norms. *Operations Research Letters* 32:510–516 2004.

[25] D. Bienstock and N. Ozbay. Computing robust basestock levels, CORC Report TR-2005-09. Columbia University, New York, 2005.

[26] J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming.* Springer, New York, 1997.

[27] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, Cambridge, UK, 2004.

[28] M. Cagetti, L. P. Hansen, T. Sargent, and N. Williams. Robust pricing with uncertain growth. *Review of Financial Studies* 15(2):363–404, 2002.

[29] H. H. Cao, T. Wang, and H. H. Zhang. Model uncertainty, limited market participation, and asset prices. *Review of Financial Studies* 18:1219–1251, 2005.

[30] X. Chen, M. Sim, and P. Sun. A robust optimization perspective of stochastic programming. Technical report, National University of Singapore, Singapore, 2004.

[31] X. Chen, M. Sim, D. Simchi-Levi, and P. Sun. Risk aversion in inventory management. Working paper, MIT, Cambridge, MA, 2004.

[32] X. Chen, M. Sim, P. Sun, and J. Zhang. A tractable approximation of stochastic programming via robust optimization. Technical report, National University of Singapore, Singapore, 2006.

[33] Z. Chen and L. G. Epstein. Ambiguity, risk, and asset returns in continuous time. *Econometrica* 70:1403–1443, 2002.

[34] M. Chou, M. Sim, and K. So. A robust framework for analyzing distribution systems with transshipment. Technical report, National University of Singapore, Singapore, 2006.

[35] L. Y. Chu, J. G. Shanthikumar, and Z. J. M. Shen. Solving operational statistics via a Bayesian analysis. Working paper, University of California, Berkeley, CA, 2005.

[36] L. Y. Chu, J. G. Shanthikumar, and Z. J. M. Shen. Pricing and revenue management with operational statistics. Working paper, University of California, Berkeley, CA, 2006.

[37] L. Y. Chu, J. G. Shanthikumar, and Z. J. M. Shen. Stochastic optimization with operational statistics: A general framework. Working paper, University of California, Berkeley, CA, 2006.

[38] S. D'Amico. Density selection and combination under model ambiguity: An application to stock returns. Technical Report 2005-09, Division of Research and Statistics and Monetary Affairs, Federal Reserve Board, Washington, D.C., 2005.

[39] X. Ding, M. L. Puterman, and A. Bisi. The censored newsvendor and the optimal acquisition of information. *Operations Research* 50:517–527, 2002.

[40] J. Dow and S. Werlang. Ambiguity aversion, risk aversion, and the optimal choice of portfolio. *Econometrica* 60:197–204, 1992.

[41] L. El Ghaoui and H. Lebret. Robust solutions to least square problems to uncertain data matrices. *SIAM Journal on Matrix Analysis and Applications* 18:1035–1064, 1997.

[42] L. El Ghaoui, F. Oustry, and H. Lebret. Robust solutions to uncertain semidefinite programs. *SIAM Journal on Optimization* 9:33–52, 1998.

[43] D. Ellsberg. Risk, ambiguity and the savage axioms. *Quarterly Journal of Economics* 75:643–669, 1961.

[44] L. G. Epstein An axiomatic model of non-Bayesian updating. *Review of Economic Studies.* Forthcoming, 2006.

[45] L. G. Epstein and J. Miao. A two-person dynamic equilibrium under ambiguity. *Journal of Economic Dynamics and Control* 27:1253–1288, 2003.

[46] L. G. Epstein and M. Schneider. Recursive multiple priors. *Journal of Economic Theory* 113:1–31, 2003.

[47] L. G. Epstein and M. Schneider. IID: Independently and indistinguishably distributed. *Journal of Economic Theory* 113:32–50, 2003.

[48] L. G. Epstein and M. Schneider. Learning under ambiguity. Working paper, University of Rochester, Rochester, NY, 2005.

[49] L. G. Epstein and M. Schneider. Ambiguity, information quality and asset pricing. Working paper, University of Rochester, Rochester, NY, 2005.

[50] L. G. Epstein and T. Wang. Intertemporal asset pricing under Knightian uncertainty. *Econometrica* 62:283–322, 1994.

[51] L. G. Epstein, J. Noor, and A. Sandroni. Non-Bayesian updating: A theoretical framework. Working paper, University of Rochester, Rochester, NY, 2005.

[52] E. Erdogan and G. Iyengar. Ambiguous chance constrained problems and robust optimization. *Mathematical Programming* 107:37–61, 2006.

[53] H. Follmer and A. Schied. Robust preferences and convex risk measures. *Advances in Finance and Stochastics*, *Essays in Honour of Dieter Sondermann.* Springer-Verlag, Berlin, Germany, 39–56, 2002.

[54] H. Follmer and A. Schied. *Stochastic Finance: An Introduction in Discrete Time. de Gruyter Studies in Mathematics* 27, 2nd ed. (2004), Berlin, Germany, 2002.

[55] G. Gallego, J. Ryan, and D. Simchi-Levi. Minimax analysis for finite horizon inventory models. *IIE Transactions* 33:861–874, 2001.

[56] L. Garlappi, R. Uppal, and T. Wang. Portfolio selection with parameter and model uncertainty: A multi-prior approach. C.E.P.R. Discussion Papers 5041, 2005.

[57] I. Gilboa and D. Schmeidler. Maxmin expected utility with non-unique prior, *Journal of Mathematical Economics* 18:141–153, 1989.

[58] D. Goldfarb and G. Iyengar. Robust portfolio selection problem. *Mathematics of Operations Research* 28:1–28, 2003.

[59] L. P. Hansen and T. J. Sargent. Acknowledging misspecification in macroeconomic theory. *Review of Economic Dynamics* 4:519–535, 2001.

[60] L. P. Hansen and T. J. Sargent. Robust control and model uncertainty. *American Economic Review* 91:60–66, 2001.

[61] L. P. Hansen and T. J. Sargent. Robust control of forward looking models. *Journal of Monetary Economics* 50(3):581–604, 2003.

[62] L. P. Hansen and T. J. Sargent. *Robustness Control and Economic Model Uncertainty.* Princeton University Press, Princeton, NJ, 2006.

[63] L. P. Hansen, T. J. Sargent, and T. D. Tallarini, Jr. Robust permanent income and pricing. *Review of Economic Studies* 66:873–907, 1999.

[64] L. P. Hansen, T. J. Sarget, and N. E. Wang. Robust permanent income and pricing with filtering. *Macroeconomic Dynamics* 6:40–84, 2002.

[65] L. P. Hansen, T. J. Sargent, G. A. Turmuhambetova, and N. Williams. Robustness and uncertainty aversion. Working paper, University of Chicago, Chicago, IL, 2002.

[66] G. Iyengar. Robust dynamic programming. *Mathematics of Operations Research* 30:257–280, 2005.

[67] A. Jain, A. E. B. Lim, and J. G. Shanthikumar. Incorporating model uncertainty and learning in operations management. Working paper, University of California Berkeley, CA, 2006.

[68] S. Karlin. Dynamic inventory policy with varying stochastic demands. *Management Science* 6:231–258, 1960.

[69] E. Kass and L. Wasserman. The selection of prior distributions by formal rules. *Journal of the American Statistical Association* 91:1343–1370, 1996.

[70] F. H. Knight. *Risk, Uncertainty and Profit.* Houghton Mifflin, Boston, MA, 1921.

[71] P. Kouvelis and G. Yu. *Robust Discrete Optimization and Its Applications.* Kluwer Academic Publishers, Boston, MA, 1997.

[72] M. A. Lariviere and E. L. Porteus. Stalking information: Bayesian inventory management with unobserved lost sales. *Management Science* 45:346–363, 1999.

[73] A. E. B. Lim and J. G. Shanthikumar. Relative entropy, exponential utility, and robust dynamic pricing. *Operations Research.* Forthcoming. 2004.

[74] A. E. B. Lim, J. G. Shanthikumar, and Z. J. M. Shen. Dynamic learning and optimization with operational statistics. Working paper, University of California, Berkeley, CA, 2006.

[75] A. E. B. Lim, J. G. Shanthikumar, and Z. J. M. Shen. Duality for relative performance objectives. Working paper, University of California, Berkeley, CA, 2006.

[76] A. E. B. Lim, J. G. Shanthikumar, and T. Watewai. Robust asset allocation with benckmarked objectives. Working paper, University of California, Berkeley, CA, 2005.

[77] A. E. B. Lim, J. G. Shanthikumar, and T. Watewai. Robust multi-product pricing. Working paper, University of California, Berkeley, CA, 2006.

[78] A. E. B. Lim, J. G. Shanthikumar, and T. Watewai. A balance between optimism and pessimism in robust portfolio choice problems through certainty equivalent ratio. Working paper, University of California, Berkeley, CA, 2006.

[79] J. Liu, J. Pan, and T. Wang. An equilibrium model of rare-event premia. *Review of Financial Studies.* Forthcoming. 2006.

[80] L. Liyanage and J. G. Shanthikumar. A practical inventory policy using operational statistics. *Operations Research Letters* 33:341–348, 2005.

[81] E. L. Porteus. *Foundations of Stochastic Inventory Theory.* Stanford University Press, Stanford, CA, 2002.

[82] C. P. Robert. *The Bayesian Choice*, 2nd ed. Springer, New York, 2001.

[83] A. Ruszczynski and A. Shapiro, eds. *Stochastic Programming. Handbooks in Operations Research and Management Series*, Vol. 10. Elsevier, New York, 2003.

[84] L. J. Savage. *The Foundations of Statistics*, 2nd ed. Dover, New York, 2003.

[85] H. Scarf. Bayes solutions of statistical inventory problem. *Annals of Mathematical Statistics* 30:490–508, 1959.

[86] A. L. Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research* 21:1154–1157, 1973.

[87] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* The MIT Press, Cambridge, MA, 1998.

[88] R. Uppal and T. Wang. Model misspecification and under diversification. *Journal of Finance* 58:2465–2486, 2003.

[89] M. H. van der Vlerk. *Stochastic Programming Bibliography.* http://mally.eco.rug.nl/spbib.html. 2006.

[90] V. N. Vapnik. *The Nature of Statistical Learning Theory*, 2nd ed. Springer, New York, 2000.

[91] A. Wald. *Statistical Decision Functions.* John Wiley and Sons, New York, 1950.

[92] P. H. Zipkin. *Foundations of Inventory Management.* McGraw Hill, New York, 2000.

TUTORIALS IN
OPERATIONS RESEARCH
INFORMS 2006

 | ISBN 13 978-1-877640-20-9
DOI 10.1287/educ.1063.0022

Robust and Data-Driven Optimization: Modern Decision Making Under Uncertainty

Dimitris Bertsimas
Sloan School of Management, Massachusetts Institute of Technology,
Cambridge, Massachusetts 02139, dbertsim@mit.edu

Aurélie Thiele
Department of Industrial and Systems Engineering, Lehigh University,
Bethlehem, Pennsylvania 18015, aurelie.thiele@lehigh.edu

Abstract Traditional models of decision making under uncertainty assume perfect information, i.e., accurate values for the system parameters and specific probability distributions for the random variables. However, such precise knowledge is rarely available in practice, and a strategy based on erroneous inputs might be infeasible or exhibit poor performance when implemented. The purpose of this tutorial is to present a mathematical framework that is well-suited to the limited information available in real-life problems and captures the decision maker's attitude toward uncertainty; the proposed approach builds on recent developments in robust and data-driven optimization. In robust optimization, random variables are modeled as uncertain parameters belonging to a convex uncertainty set, and the decision maker protects the system against the worst case within that set. Data-driven optimization uses observations of the random variables as direct inputs to the mathematical programming problems. The first part of the tutorial describes the robust optimization paradigm in detail in single-stage and multistage problems. In the second part, we address the issue of constructing uncertainty sets using historical realizations of the random variables and investigate the connection between convex sets, in particular polyhedra, and a specific class of risk measures.

Keywords optimization under uncertainty; risk preferences; uncertainty sets; linear programming

1. Introduction

The field of decision making under uncertainty was pioneered in the 1950s by Charnes and Cooper [23] and Dantzig [25], who set the foundation for, respectively, stochastic programming and optimization under probabilistic constraints. While these classes of problems require very different models and solution techniques, they share the same assumption that the probability distributions of the random variables are known exactly, and despite Scarf's [38] early observation that "we may have reason to suspect that the future demand will come from a distribution that differs from that governing past history in an unpredictable way," most research efforts in decision making under uncertainty over the past decades have relied on the precise knowledge of the underlying probabilities. Even under this simplifying assumption, a number of computational issues arises, e.g., the need for multivariate integration to evaluate chance constraints and the large-scale nature of stochastic programming problems. The reader is referred to Birge and Louveaux [22] and Kall and Mayer [31] for an overview of solution techniques. Today, stochastic programming has established itself as a powerful modeling tool when an accurate probabilistic description of the randomness is available; however, in many real-life applications the decision maker does not have this

information—for instance, when it comes to assessing customer demand for a product. (The lack of historical data for new items is an obvious challenge to estimating probabilities, but even well-established product lines can face sudden changes in demand due to the market entry by a competitor or negative publicity.) Estimation errors have notoriously dire consequences in industries with long production lead times such as automotive, retail, and high-tech, where they result in stockpiles of unneeded inventory or, at the other end of the spectrum, lost sales and customers' dissatisfaction. The need for an alternative, nonprobabilistic theory of decision making under uncertainty has become pressing in recent years because of volatile customer tastes, technological innovation, and reduced product life cycles, which reduce the amount of information available and make it obsolete faster.

In mathematical terms, imperfect information threatens the relevance of the solution obtained by the computer in two important aspects: (i) the solution might not actually be feasible when the decision maker attempts to implement it, and (ii) the solution, when feasible, might lead to a far greater cost (or smaller revenue) than the truly optimal strategy. Potential infeasibility, e.g., from errors in estimating the problem parameters, is the primary concern of the decision maker. The field of operations research remained essentially silent on that issue until Soyster's work [44], where every uncertain parameter in convex programming problems was taken equal to its worst-case value within a set. While this achieved the desired effect of immunizing the problem against parameter uncertainty, it was widely deemed too conservative for practical implementation. In the mid-1990s, research teams led by Ben-Tal and Nemirovski [4, 5, 6], El-Ghaoui and Lebret [27], and El-Ghaoui et al. [28] addressed the issue of overconservatism by restricting the uncertain parameters to belong to ellipsoidal uncertainty sets, which removes the most unlikely outcomes from consideration and yields tractable mathematical programming problems. In line with these authors' terminology, optimization for the worst-case value of parameters within a set has become known as "robust optimization." A drawback of the robust modeling framework with ellipsoidal uncertainty sets is that it increases the complexity of the problem considered, e.g., the robust counterpart of a linear programming problem is a second-order cone problem. More recently, Bertsimas et al. [20] and Bertsimas and Sim [14, 15] have proposed a robust optimization approach based on polyhedral uncertainty sets, which preserves the class of problems under analysis—e.g., the robust counterpart of a linear programming problem remains a linear programming problem—and thus has advantages in terms of tractability in large-scale settings. It can also be connected to the decision maker's attitude toward uncertainty, providing guidelines to construct the uncertainty set from the historical realizations of the random variables using data-driven optimization (Bertsimas and Brown [12]).

The purpose of this tutorial is to illustrate the capabilities of the robust, data-driven optimization framework as a modeling tool in decision making under uncertainty, and, in particular, to

(1) Address estimation errors of the problem parameters and model random variables in single-stage settings (§2),

(2) Develop a tractable approach to dynamic decision making under uncertainty, incorporating that information is revealed in stages (§3), and

(3) Connect the decision maker's risk preferences with the choice of uncertainty set using the available data (§4).

2. Static Decision Making Under Uncertainty

2.1. Uncertainty Model

In this section, we present the robust optimization framework when the decision maker must select a strategy before (or without) knowing the exact value taken by the uncertain parameters. Uncertainty can take two forms: (i) estimation errors for parameters of constant but unknown value, and (ii) stochasticity of random variables. The model here does not

allow for recourse, i.e, remedial action once the values of the random variables become known. Section 3 addresses the case where the decision maker can adjust his strategy to the information revealed over time.

Robust optimization builds on the following two principles, which have been identified by Nahmias [32], Sheffi [41], and Simchi-Levi et al. [43] as fundamental to the practice of modern operations management under uncertainty:

- Point forecasts are meaningless (because they are always wrong) and should be replaced by range forecasts.
- Aggregate forecasts are more accurate than individual ones.

The framework of robust optimization incorporates these managerial insights into quantitative decision models as follows. We model uncertain quantities (parameters or random variables) as parameters belonging to a prespecified interval—the *range forecast*—provided for instance by the marketing department. Such forecasts are in general symmetric around the point forecast, i.e., the nominal value of the parameter considered. The greater accuracy of aggregate forecasting will be incorporated by an *additional constraint* limiting the maximum deviation of the aggregate forecast from its nominal value.

To present the robust framework in mathematical terms, we follow closely Bertsimas and Sim [15] and consider the linear programming problem:

$$\begin{aligned} \min\ & \mathbf{c}'\mathbf{x} \\ \text{s.t.}\ & \mathbf{A}\mathbf{x} \geq \mathbf{b}, \\ & \mathbf{x} \in X, \end{aligned} \tag{1}$$

where uncertainty is assumed without loss of generality to affect only the constraint coefficients, $\mathbf{A}$, and X is a polyhedron (not subject to uncertainty). Problem (1) arises in a wide range of settings; it can, for instance, be interpreted as a *production planning problem* in which the decision maker must purchase raw material to minimize cost while meeting the demand for each product, despite uncertainty on the machine productivities. Note that a problem with uncertainty in the cost vector $\mathbf{c}$ and the right side of $\mathbf{b}$ can immediately be reformulated as

$$\begin{aligned} \min\ & Z \\ \text{s.t.}\ & Z - \mathbf{c}'\mathbf{x} \geq 0, \\ & \mathbf{A}\mathbf{x} - \mathbf{b}y \geq \mathbf{0}, \\ & \mathbf{x} \in X, \quad y = 1, \end{aligned} \tag{2}$$

which has the form of problem (1).

The fundamental issue in problem (1) is one of *feasibility*; in particular, the decision maker will guarantee that every constraint is satisfied for any possible value of $\mathbf{A}$ in a given convex uncertainty set $\mathcal{A}$ (which will be described in detail shortly). This leads to the following formulation of the *robust counterpart* of problem (1):

$$\begin{aligned} \min\ & \mathbf{c}'\mathbf{x} \\ \text{s.t.}\ & \mathbf{a}_i'\mathbf{x} \geq b_i, \quad \forall i,\ \forall \mathbf{a}_i \in \mathcal{A}, \\ & \mathbf{x} \in X, \end{aligned} \tag{3}$$

or equivalently:

$$\begin{aligned} \min\ & \mathbf{c}'\mathbf{x} \\ \text{s.t.}\ & \min_{\mathbf{a}_i \in \mathcal{A}} \mathbf{a}_i'\mathbf{x} \geq b_i, \quad \forall i, \\ & \mathbf{x} \in X, \end{aligned} \tag{4}$$

where $\mathbf{a}_i$ is the ith vector of $\mathbf{A}'$.

Solving the robust problem as it is formulated in problem (4) would require evaluating $\min_{\mathbf{a}_i \in \mathcal{A}} \mathbf{a}_i' \mathbf{x}$ for each candidate solution $\mathbf{x}$, which would make the robust formulation considerably more difficult to solve than its nominal counterpart, a linear programming problem. The key insight that preserves the computational tractability of the robust approach is that problem (4) can be reformulated as a single convex programming problem for any convex uncertainty set $\mathcal{A}$, and specifically, a linear programming problem when $\mathcal{A}$ is a polyhedron (see Ben-Tal and Nemirovski [5]). We now justify this insight by describing the construction of a *tractable, linear* equivalent formulation of problem (4).

The set $\mathcal{A}$ is defined as follows. To simplify the exposition, we assume that every coefficient a_{ij} of the matrix $\mathbf{A}$ is subject to uncertainty, and that all coefficients are independent. The decision maker knows range forecasts for all the uncertain parameters, specifically, parameter a_{ij} belongs to a symmetric interval $[\bar{a}_{ij} - \hat{a}_{ij}, \bar{a}_{ij} + \hat{a}_{ij}]$ centered at the point forecast $\bar{a}_{ij}$. The half-length $\hat{a}_{ij}$ measures the precision of the estimate. We define the *scaled deviation* z_{ij} of parameter a_{ij} from its nominal value as

$$z_{ij} = \frac{a_{ij} - \bar{a}_{ij}}{\hat{a}_{ij}}. \tag{5}$$

The scaled deviation of a parameter always belongs to $[-1, 1]$.

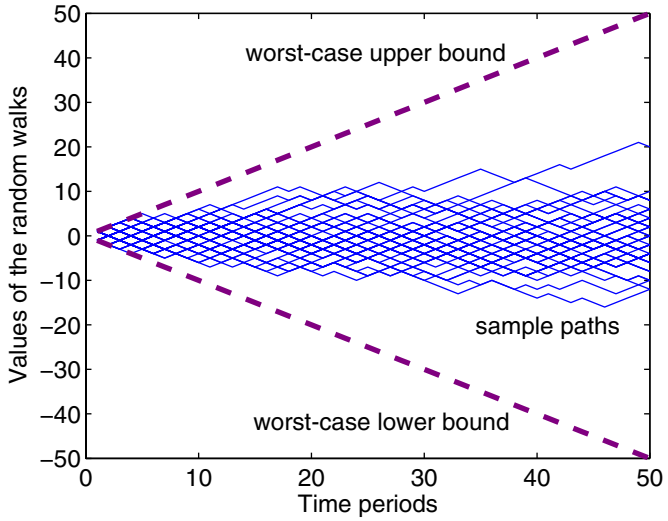
Although the aggregate scaled deviation for constraint i, $\sum_{j=1}^{n} z_{ij}$, could in theory take any value between $-n$ and n, because aggregate forecasts are more accurate than individual ones suggests that the "true values" taken by $\sum_{j=1}^{n} z_{ij}$ will belong to a much narrower range. Intuitively, some parameters will exceed their point forecast while others will fall below estimate, so the z_{ij} will tend to cancel each other out. This is illustrated in Figure 1, where we have plotted 50 sample paths of a symmetric random walk over 50 time periods. Figure 1 shows that, when there are few sources of uncertainty (few time periods, little aggregation), the random walk might indeed take its worst-case value; however, as the number of sources of uncertainty increases, this becomes extremely unlikely, as evidenced by the concentration of the sample paths around the mean value of 0.

We incorporate this point in mathematical terms as

$$\sum_{j=1}^{n} |z_{ij}| \le \Gamma_i, \quad \forall i. \tag{6}$$

The parameter Γ_i, which belongs to $[0, n]$, is called the *budget of uncertainty* of constraint i. If Γ_i is integer, it is interpreted as the maximum number of parameters that can deviate

FIGURE 1. Sample paths as a function of the number of random parameters.



from their nominal values.

- If $\Gamma_i = 0$, the z_{ij} for all j are forced to 0, so that the parameters a_{ij} are equal to their point forecasts $\bar{a}_{ij}$ for all j, and there is no protection against uncertainty.
- If $\Gamma_i = n$, constraint (6) is redundant with the fact that $|z_{ij}| \le 1$ for all j. The ith constraint of the problem is completely protected against uncertainty, which yields a very conservative solution.
- If $\Gamma_i \in (0, n)$, the decision maker makes a trade-off between the protection level of the constraint and the degree of conservatism of the solution.

We provide guidelines to select the budgets of uncertainty at the end of this section. The set $\mathcal{A}$ becomes

$$\mathcal{A} = \{(a_{ij}) \mid a_{ij} = \bar{a}_{ij} + \hat{a}_{ij} z_{ij}, \forall i, j,\ \mathbf{z} \in \mathcal{Z}\}. \tag{7}$$

with

$$\mathcal{Z} = \left\{ \mathbf{z} \;\middle|\; |z_{ij}| \le 1, \forall i, j, \sum_{j=1}^{n} |z_{ij}| \le \Gamma_i, \forall i \right\}, \tag{8}$$

and problem (4) can be reformulated as

$$\begin{aligned} \min\ & \mathbf{c}'\mathbf{x} \\ \text{s.t. } & \bar{\mathbf{a}}_i'\mathbf{x} + \min_{z_i \in \mathcal{Z}_i} \sum_{j=1}^{n} \hat{a}_{ij} x_j z_{ij} \ge b_i, \quad \forall i, \\ & \mathbf{x} \in X, \end{aligned} \tag{9}$$

where $\mathbf{z}_i$ is the vector whose jth element is z_{ij} and $\mathcal{Z}_i$ is defined as

$$\mathcal{Z}_i = \left\{ \mathbf{z}_i \;\middle|\; |z_{ij}| \le 1, \forall j, \sum_{j=1}^{n} |z_{ij}| \le \Gamma_i \right\}. \tag{10}$$

$\min_{z_i \in \mathcal{Z}_i} \sum_{j=1}^{n} \hat{a}_{ij} x_j z_{ij}$ for a given i is equivalent to

$$\begin{aligned} -\max\ & \sum_{j=1}^{n} \hat{a}_{ij} |x_j| z_{ij} \\ \text{s.t. } & \sum_{j=1}^{n} z_{ij} \le \Gamma_i, \\ & 0 \le z_{ij} \le 1, \quad \forall j, \end{aligned} \tag{11}$$

which is linear in the decision vector $\mathbf{z}_i$. Applying strong duality arguments to problem (11) (see Bertsimas and Sim [15] for details), we then reformulate the robust problem as a *linear programming problem*:

$$\begin{aligned} \min\ & \mathbf{c}'\mathbf{x} \\ \text{s.t. } & \bar{\mathbf{a}}_i'\mathbf{x} - \Gamma_i p_i - \sum_{j=1}^{n} q_{ij} \ge b_i, \quad \forall i, \\ & p_i + q_{ij} \ge \hat{a}_{ij} y_j, \quad \forall i, j, \\ & -y_j \le x_j \le y_j, \quad \forall j, \\ & p_i, q_{ij} \ge 0, \quad \forall i, j, \\ & \mathbf{x} \in X. \end{aligned} \tag{12}$$

With m the number of constraints subject to uncertainty and n the number of variables in the deterministic problem (1), problem (12) has $n + m(n+1)$ new variables and $n(m+2)$ new constraints besides nonnegativity. An appealing feature of this formulation is that linear

programming problems can be solved efficiently, including by the commercial software used in industry.

At optimality,

(1) y_j will equal $|x_j|$ for any j,

(2) p_i will equal the $\lceil \Gamma_i \rceil$-th greatest $\hat{a}_{ij}|x_j|$, for any i,

(3) q_{ij} will equal $\hat{a}_{ij}|x_j| - p_i$ if $\hat{a}_{ij}|x_j|$ is among the $\lfloor \Gamma_i \rfloor$-th greatest $\hat{a}_{ik}|x_k|$ and 0 otherwise, for any i and j. (Equivalently, $q_{ij} = \max(0, \hat{a}_{ij}|x_j| - p_i)$.)

To implement this framework, the decision maker must now assign a value to the budget of uncertainty Γ_i for each i. The values of the budgets can, for instance, reflect the manager's own attitude toward uncertainty; the connection between risk preferences and uncertainty sets is studied in depth in §4. Here, we focus on selecting the budgets so that the constraints $\mathbf{Ax} \geq \mathbf{b}$ are satisfied with high probability in practice, despite the lack of precise information on the distribution of the random matrix $\mathbf{A}$. The central result linking the value of the budget to the probability of constraint violation is due to Bertsimas and Sim [15] and can be summarized as follows:

For the constraint $\mathbf{a}_i'\mathbf{x} \geq b_i$ to be violated with probability at most ϵ_i, when each a_{ij} obeys a symmetric distribution centered at $\bar{a}_{ij}$ and of support $[\bar{a}_{ij} - \hat{a}_{ij}, \bar{a}_{ij} + \hat{a}_{ij}]$, it is sufficient to choose Γ_i at least equal to $1 + \Phi^{-1}(1 - \epsilon_i)\sqrt{n}$, where Φ is the cumulative distribution of the standard Gaussian random variable.

As an example, for $n = 100$ sources of uncertainty and $\epsilon_i = 0.05$ in constraint i, Γ_i must be at least equal to 17.4, i.e., it is sufficient to protect the system against only 18% of the uncertain parameters taking their worst-case value. Most importantly, Γ_i is always of the order of $\sqrt{n}$. Therefore, the constraint can be protected with high probability while keeping the budget of uncertainty, and hence the degree of conservatism of the solution, moderate.

We now illustrate the approach on a few simple examples.

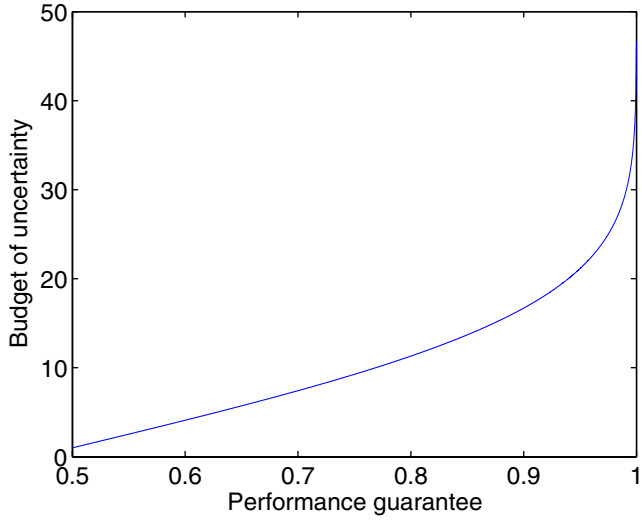
Example 2.1. Portfolio Management (Bertsimas and Sim [15]). A decision maker must allocate her/his wealth among 150 assets in to maximize his return. S/he has established that the return of asset i belongs to the interval $[r_i - s_i, r_i + s_i]$ with $r_i = 1.15 + i(0.05/150)$ and $s_i = (0.05/450)\sqrt{300 \cdot 151 \cdot i}$. Short sales are not allowed. Obviously, in the deterministic problem in which all returns are equal to their point forecasts, it is optimal to invest everything in the asset with the greatest nominal return, here, asset 150. (Similarly, in the conservative approach in which all returns equal their worst-case values, it is optimal to invest everything in the asset with the greatest worst-case return, which is asset 1.)

Figure 2 depicts the minimum budget of uncertainty required to guarantee an appropriate performance for the investor, in this context meaning that the actual value of his portfolio will exceed the value predicted by the robust optimization model with probability at least equal to the numbers on the x-axis. We note that performance requirements of up to 98% can be achieved by a small budget of uncertainty ($\Gamma \approx 26$, protecting about 17% of the sources of randomness), but more-stringent constraints require a drastic increase in the protection level, as evidenced by the almost vertical increase in the curve.

The investor would like to find a portfolio allocation such that there is only a probability of 5% that the actual portfolio value will fall below the value predicted by her/his optimization model. Therefore, s/he picks $\Gamma \geq 21.15$, e.g., $\Gamma = 22$, and solves the linear programming problem:

$$\begin{aligned} \max \ & \sum_{i=1}^{150} r_i x_i - \Gamma p - \sum_{i=1}^{150} q_i \\ \text{s.t.} \ & \sum_{i=1}^{150} x_i = 1, \\ & p + q_i \geq s_i x_i, \quad \forall\, i, \\ & p, q_i, x_i \geq 0, \quad \forall\, i. \end{aligned} \tag{13}$$

FIGURE 2. Minimum budget of uncertainty to ensure performance guarantee.



At optimality, he invests in every asset, and the fraction of wealth invested in asset i decreases from 4.33% to 0.36% as the index i increases from 1 to 150. The optimal objective is 1.1452.

To illustrate the impact of the robust methodology, assume the true distribution of the return of asset i is Gaussian with mean r_i and standard deviation $s_i/2$, so that the range forecast for return i includes every value within two standard deviations of the mean. Asset returns are assumed to be independent.

- The portfolio value in the *nominal* strategy, where everything is invested in asset 150, obeys a Gaussian distribution with mean 1.2 and standard deviation 0.1448.
- The portfolio value in the *conservative* strategy, where everything is invested in asset 1, obeys a Gaussian distribution with mean 1.1503 and standard deviation 0.0118.
- The portfolio value in the *robust* strategy, which leads to a diversification of the investor's holdings, obeys a Gaussian distribution with mean 1.1678 and standard deviation 0.0063.

Hence, not taking uncertainty into account rather than implementing the robust strategy increases risk (measured by the standard deviation) by a factor of 23 while yielding an increase in expected return of only 2.7%, and being too pessimistic regarding the outcomes doubles the risk and also decreases the expected return.

Example 2.2. Inventory Management (Thiele [45]). A warehouse manager must decide how many products to order, given that the warehouse supplies n stores and it is only possible to order once for the whole planning period. The warehouse has an initial inventory of zero, and incurs a unit shortage cost s per unfilled item and a unit holding cost h per item remaining in the warehouse at the end of the period. Store demands are assumed to be i.i.d. with a symmetric distribution around the mean, and all stores have the same range forecast $[\bar{w}-\hat{w}, \bar{w}+\hat{w}]$ with $\bar{w}$ the nominal forecast, common to each store. Let x be the number of items ordered by the decision maker, whose goal is to minimize the total cost $\max\{h(x-\sum_{i=1}^n w_i), s(\sum_{i=1}^n w_i - x)\}$, with $\sum_{i=1}^n w_i$ the actual aggregate demand. The robust problem for a given budget of uncertainty Γ can be formulated as

$$\begin{aligned} \min\ & Z \\ \text{s.t.}\ & Z \geq h(x - n\bar{w} + \Gamma\hat{w}), \\ & Z \geq s(-x + n\bar{w} + \Gamma\hat{w}), \\ & x \geq 0. \end{aligned} \tag{14}$$

The solution to problem (14) is available in closed form and is equal to

$$x_\Gamma = n\bar{w} + \frac{s-h}{s+h}\Gamma\hat{w}. \tag{15}$$

The optimal objective is then

$$C_\Gamma = \frac{2hs}{s+h}\Gamma\hat{w}. \tag{16}$$

If shortage is more penalized than holding, the decision maker will order more than the nominal aggregate forecast, and the excess amount will be proportional to the maximum deviation $\Gamma\hat{w}$, as well as the ratio $(s-h)/(s+h)$. The optimal order is linear in the budget of uncertainty.

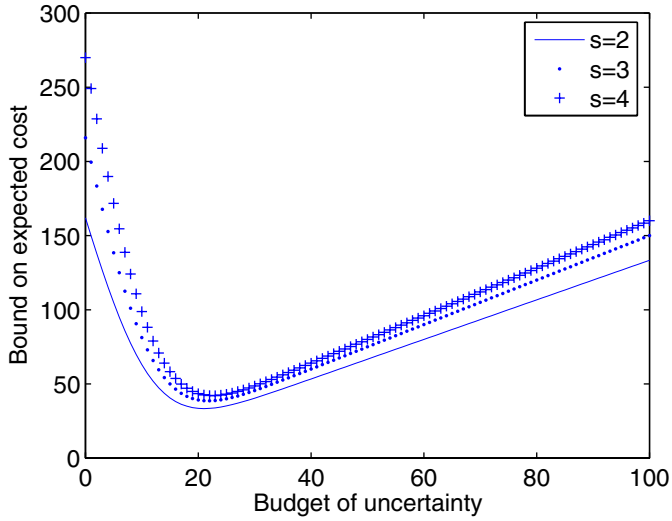
Using the central limit theorem, and assuming that the variance of each store demand is known and equal to σ^2, it is straightforward to show that the optimal objective C_Γ is an upper bound to the true cost with probability $1-\epsilon$ when Γ is at least equal to $(\sigma/\hat{w})\sqrt{n}\Phi^{-1}(1-\epsilon/2)$. This formula is independent of the cost parameters h and s. For instance, with $n=100$ and $\hat{w}=2\sigma$, the actual cost falls below C_{10} with probability 0.95.

Because, in this case, the optimal solution is available in closed form, we can analyze in more depth the impact of the budget of uncertainty on the practical performance of the robust solution. To illustrate the two dangers of "not worrying enough" about uncertainty (i.e., only considering the nominal values of the parameters) and "worrying too much" (i.e., only considering their worst-case values) in practical implementations, we compute the expected cost for the worst-case probability distribution of the aggregate demand W. We only use the following information on W: its distribution is symmetric with mean $n\bar{w}$ and support $[n(\bar{w}-\hat{w}), n(\bar{w}+\hat{w})]$, and (as established by Bertsimas and Sim [15]) W falls within $[n\bar{w}-\Gamma\hat{w}, n\bar{w}+\Gamma\hat{w}]$ with probability $2\phi-1$ where $\phi=\Phi((\Gamma-1)/\sqrt{n})$. Let $\mathcal{W}$ be the set of probability distributions satisfying these assumptions. Thiele [45] proves the following bound:

$$\max_{W\in\mathcal{W}} E[\max\{h(x-W), s(W-x)\}] = \hat{w}(s+h)\left[n(1-\phi)+\Gamma\left\{\phi - \frac{s^2+h^2}{(s+h)^2}\right\}\right]. \tag{17}$$

In Figure 3, we plot this upper bound on the expected cost for $n=100$, $\hat{w}=1$, $h=1$, and $s=2$, 3, and 4. We note that not incorporating uncertainty in the model is the more costly mistake the manager can make in this setting (as opposed to being too conservative), the penalty increases when the shortage cost increases. The budget of uncertainty minimizing

FIGURE 3. Maximum expected cost as a function of the budget of uncertainty.



this bound is approximately equal to 20 and does not appear to be sensitive to the value of the cost parameters.

The key insight of Figure 3 is that accounting for a *limited* amount of uncertainty via the robust optimization framework leads to significant cost benefits. A decision maker implementing the nominal strategy will be penalized for not planning at all for randomness—i.e., the aggregate demand deviating from its point forecast—but protecting the system against the most negative outcome will also result in lost profit opportunities. The robust optimization approach achieves a trade-off between these two extremes.

2.2. Extensions

2.2.1. Discrete Decision Variables. The modeling power of robust optimization also extends to discrete decision variables. Integer decision variables can be incorporated into the set X (which is then no longer a polyhedron), while binary variables allow for the development of a specifically tailored algorithm due to Bertsimas and Sim [14]. We describe this approach for the binary programming problem:

$$\begin{aligned} \max\ & \mathbf{c}'\mathbf{x} \\ \text{s.t.}\ & \mathbf{a}'\mathbf{x} \le b \\ & \mathbf{x} \in \{0,1\}^n. \end{aligned} \tag{18}$$

Problem (18) can be interpreted as a *capital allocation problem* in which the decision maker must choose between n projects to maximize her/his payoff under a budget constraint, but does not know exactly how much money each project will require. In this setting, the robust problem (12) (modified to take into account the sign of the inequality and the maximization) becomes

$$\begin{aligned} \max\ & \mathbf{c}'\mathbf{x} \\ \text{s.t.}\ & \bar{\mathbf{a}}'\mathbf{x} + \Gamma p + \sum_{j=1}^{n} q_j \le b \\ & p + q_j \ge \hat{a}_j x_j, \quad \forall j, \\ & p \ge 0, \quad \mathbf{q} \ge \mathbf{0}, \\ & \mathbf{x} \in \{0,1\}^n. \end{aligned} \tag{19}$$

As noted for problem (12), at optimality, q_j will equal $\max(0, \hat{a}_j x_j - p)$. The major insight here is that, because x_j is binary, q_j can take only two values—$\max(0, \hat{a}_j - p)$ and 0—which can be rewritten as $\max(0, \hat{a}_j - p)x_j$. Therefore, the optimal p will be one of the $\hat{a}_j$, and the optimal solution can be found by solving n subproblems *of the same size and structure* as the original deterministic problem, and keeping the one with the highest objective. Solving these subproblems can be automated with no difficulty, for instance, in AMPL/CPLEX, thus preserving the computational tractability of the robust optimization approach. Subproblem i, $i = 1, \ldots, n$, is defined as the following *binary programming problem*:

$$\begin{aligned} \max\ & \mathbf{c}'\mathbf{x} \\ \text{s.t.}\ & \bar{\mathbf{a}}'\mathbf{x} + \sum_{j=1}^{n} \max(0, \hat{a}_j - \hat{a}_i) x_j \le b - \Gamma \hat{a}_i \\ & \mathbf{x} \in \{0,1\}^n. \end{aligned} \tag{20}$$

It has the same number of constraints and decision variables as the original problem.

Example 2.3. Capital Allocation (Bertsimas and Sim [14]). The manager has a budget b of \$4,000 and can choose among 200 projects. The nominal amount of money $\bar{a}_i$ required to complete project i is chosen randomly from the set $\{20, \ldots, 29\}$, the range

forecast allows for a deviation of at most 10% of this estimate. The value (or importance) c_i of project i is chosen randomly from $\{16, \ldots, 77\}$. Bertsimas and Sim [14] show that, while the nominal problem yields an optimal value of 5,592, taking Γ equal to 37 ensures that the decision maker will remain within budget with a probability of 0.995, and with a decrease in the objective value of only 1.5%. Therefore, the system can be protected against uncertainty at very little cost.

2.2.2. Generic Polyhedral Uncertainty Sets and Norms. Because the main mathematical tool used in deriving tractable robust formulations is the use of strong duality in linear programming, it should not be surprising that the robust counterparts to linear problems with generic polyhedral uncertainty sets remain linear. For instance, if the set $\mathcal{Z}_i$ for constraint i is defined by $\mathcal{Z}_i = \{\mathbf{z} \mid \mathbf{F}_i|\mathbf{z}| \leq \mathbf{g}_i, |\mathbf{z}| \leq \mathbf{e}\}$ where $\mathbf{e}$ is the unit vector, rather than $\mathcal{Z}_i = \{\mathbf{z} \mid \sum_{j=1}^{n_i} |z_{ij}| \leq \Gamma_i, |z_{ij}| \leq 1, \forall j\}$, it is immediately possible to formulate the robust problem as

$$\begin{aligned}
\min\ & \mathbf{c}'\mathbf{x} \\
\text{s.t.}\ & \bar{\mathbf{a}}_i'\mathbf{x} - \mathbf{g}_i'\mathbf{p}_i - \mathbf{e}'\mathbf{q}_i \geq b_i, \quad \forall i, \\
& \mathbf{F}_i'\mathbf{p}_i + \mathbf{q}_i \geq (\operatorname{diag} \hat{\mathbf{a}}_i)\mathbf{y}, \quad \forall i, \\
& -\mathbf{y} \leq \mathbf{x} \leq \mathbf{y}, \\
& \mathbf{p}, \mathbf{q} \geq \mathbf{0}, \\
& \mathbf{x} \in X. \qquad (21)
\end{aligned}$$

Moreover, given that the precision of each individual forecast $\bar{a}_{ij}$ is quantified by the parameter $\hat{a}_{ij}$, which measures the maximum "distance" of the true scalar parameter a_{ij} from its nominal value $\bar{a}_{ij}$, it is natural to take this analysis one step further and consider the distance of the true vector of parameters $\mathbf{A}$ from its point forecast $\widetilde{A}$. Uncertainty sets arising from limitations on the distance (measured by an arbitrary norm) between uncertain coefficients and their nominal values have been investigated by Bertsimas et al. [20], who show that reframing the uncertainty set in those terms leads to convex problems with constraints involving a dual norm, and provide a unified treatment of robust optimization as described by Ben-Tal and Nemirovski [4, 5], Bertsimas and Sim [15], El-Ghaoui and Lebret [27], and El-Ghaoui et al. [28]. Intuitively, robust optimization protects the system against any value of the parameter vector within a prespecified "distance" from its point forecast.

2.2.3. Additional Models and Applications. Robust optimization has been at the center of many research efforts over the last decade, and in this last paragraph we mention a few of those pertaining to static decision making under uncertainty for the interested reader. This is, of course, far from an exhaustive list.

While this tutorial focuses on linear programming and polyhedral uncertainty sets, the robust optimization paradigm is well suited to a much broader range of problems. Atamturk [2] provides strong formulations for robust mixed 0-1 programming under uncertainty in the objective coefficients. Sim [42] extends the robust framework to quadratically constrained quadratic problems, conic problems as well as semidefinite problems, and provides performance guarantees. Ben-Tal et al. [8] consider tractable approximations to robust conic-quadratic problems. An important application area is portfolio management, in which Goldfarb and Iyengar [29] protect the optimal asset allocation from estimation errors in the parameters by using robust optimization techniques. Ordonez and Zhao [34] apply the robust framework to the problem of expanding network capacity when demand and travel times are uncertain. Finally, Ben-Tal et al. [7] investigate robust problems in which the decision maker requires a controlled deterioration of the performance when the data falls outside the uncertainty set.

3. Dynamic Decision Making Under Uncertainty

3.1. Generalities

Section 2 has established the power of robust optimization in static decision making, where it immunizes the solution against infeasibility and suboptimality. We now extend our presentation to the dynamic case. In this setting, information is revealed *sequentially* over time, and the manager makes a *series of decisions*, which takes into account the historical realizations of the random variables. Because dynamic optimization involves multiple decision epochs and must capture the wide range of circumstances (i.e., state of the system, values taken by past sources of randomness) in which decisions are made, the fundamental issue here is one of *computational tractability*.

Multistage stochastic models provide an elegant theoretical framework to incorporate uncertainty revealed over time (see Bertsekas [11] for an introduction). However, the resulting large-scale formulations quickly become intractable as the size of the problem increases, thus limiting the practical usefulness of these techniques. For instance, a manager planning for the next quarter (13 weeks) and considering three values of the demand each week (high, low, or medium) has just created $3^{13} \approx 1.6$ million scenarios in the stochastic framework. Approximation schemes such as neurodynamic programming (Bertsekas and Tsitsiklis [18]) have yet to be widely implemented, in part because of the difficulty in finetuning the approximation parameters. Moreover, as in the static case, each scenario needs to be assigned a specific probability of occurrence, and the difficulty in estimating these parameters accurately is compounded in multistage problems by long time horizons. Intuitively, "one can predict tomorrow's value of the Dow Jones Industrial Average more accurately than next year's value" (Nahmias [32]).

Therefore, a decision maker using a stochastic approach might expand considerable computational resources to solve a multistage problem, which will *not* be the true problem s/he is confronted with because of estimation errors. A number of researchers have attempted to address this issue by implementing robust techniques directly in the stochastic framework (i.e., optimizing over the worst-case probabilities in a set), e.g., Dupačová [26], Shapiro [40], and Žáčková [48] for two-stage stochastic programming, and Iyengar [30] and Nilim and El-Ghaoui [33] for multistage dynamic programming. Although this method protects the system against parameter ambiguity, it suffers from the same limitations as the algorithm with perfect information; hence, if a problem relying on a probabilistic description of the uncertainty is computationally intractable, its robust counterpart will be intractable as well.

In contrast, we approach dynamic optimization problems subject to uncertainty by representing the *random variables*, rather than the underlying probabilities, as uncertain parameters belonging to given uncertainty sets. This is in line with the methodology presented in the static case. The extension of the approach to dynamic environments raises the following questions:

(1) Is the robust optimization paradigm *tractable* in dynamic settings?

(2) Does the manager derive *deeper insights* into the impact of uncertainty?

(3) Can the methodology incorporate the *additional information* received by the decision maker over time?

As explained below, the answer to each of these three questions is *yes*.

3.2. A First Model

A first, intuitive approach is to incorporate uncertainty to the underlying *deterministic* formulation. In this tutorial, we focus on applications that can be modeled (or approximated) as linear programming problems when there is no randomness. For clarity, we present the framework in the context of inventory management; the exposition closely follows Bertsimas and Thiele [17].

3.2.1. Scalar Case. We start with the simple case where the decision maker must decide how many items to order at each time period at a single store. (In mathematical terms, the state of the system can be described as a scalar variable, specifically, the amount of inventory in the store.) We use the following notation.

x_t: inventory at the beginning of time period t
u_t: amount ordered at the beginning of time period t
w_t: demand occurring during time period t

Demand is backlogged over time, and orders made at the beginning of a time period arrive at the end of that same period. Therefore, the dynamics of the system can be described as a *linear* equation

$$x_{t+1} = x_t + u_t - w_t, \tag{22}$$

which yields the closed-form formula

$$x_{t+1} = x_0 + \sum_{\tau=0}^{t} (u_\tau - w_\tau). \tag{23}$$

The cost incurred at each time period has two components:

(1) An ordering cost linear in the amount ordered, with c the unit ordering cost (Bertsimas and Thiele [17] also consider the case of a fixed cost charged whenever an order is made), and

(2) An inventory cost, with h, respectively s, the unit cost charged per item held in inventory, respectively backlogged, at the end of each time period.

The decision maker seeks to minimize the total cost over a time horizon of length T. S/he has a range forecast $[\bar{w}_t - \hat{w}_t, \bar{w}_t + \hat{w}_t]$, centered at the nominal forecast $\bar{w}_t$, for the demand at each time period t, with $t = 0, \ldots, T-1$. If there is no uncertainty, the problem faced by the decision maker can be formulated as a linear programming problem:

$$\begin{aligned} \min\ & c \sum_{t=0}^{T-1} u_t + \sum_{t=0}^{T-1} y_t \\ \text{s.t.}\ & y_t \geq h\left(x_0 + \sum_{\tau=0}^{t} (u_\tau - \bar{w}_\tau) \right), \quad \forall t \\ & y_t \geq -s\left(x_0 + \sum_{\tau=0}^{t} (u_\tau - \bar{w}_\tau) \right), \quad \forall t, \\ & u_t \geq 0, \quad \forall t. \end{aligned} \tag{24}$$

At optimality, y_t is equal to the inventory cost computed at the end of time period t, i.e., $\max(hx_{t+1}, -sx_{t+1})$. The optimal solution to problem (24) is to order nothing if there is enough in inventory at the beginning of period t to meet the demand $\bar{w}_t$ and order the missing items, i.e., $\bar{w}_t - x_t$, otherwise, which is known in inventory management as an (S, S) *policy* with basestock level $\bar{w}_t$ at time t. (The basestock level quantifies the amount of inventory on hand or on order at a given time period, see Porteus [35].)

The robust optimization approach consists in replacing each deterministic demand $\bar{w}_t$ by an uncertain parameter $w_t = \bar{w}_t + \hat{w}_t z_t$, $|z_t| \leq 1$, for all t, and guaranteeing that the constraints hold for any scaled deviations belonging to a given uncertainty set. Because the constraints depend on the time period, the uncertainty set will depend on the time period as well and, specifically, the amount of uncertainty faced by the cumulative demand up to (and including) time t. This motivates introducing a *sequence* of budgets of uncertainty Γ_t, $t = 0, \ldots, T-1$, rather than using a single budget as in the static case. Natural requirements for such a sequence are that the budgets increase over time, as uncertainty increases with

the length of the time horizon considered, and do not increase by more than one at each time period, because only one new source of uncertainty is revealed at any time.

Let $\bar{x}_t$ be the amount in inventory at time t if there is no uncertainty: $\bar{x}_{t+1} = x_0 + \sum_{\tau=0}^{t}(u_\tau - \bar{w}_\tau)$ for all t. Also, let Z_t be the optimal solution of

$$\begin{aligned} \max \ & \sum_{\tau=0}^{t} \hat{w}_\tau z_\tau \\ \text{s.t.} \ & \sum_{\tau=0}^{t} z_\tau \le \Gamma_t, \\ & 0 \le z_\tau \le 1, \quad \forall \tau \le t. \end{aligned} \tag{25}$$

From $0 \le \Gamma_t - \Gamma_{t-1} \le 1$, it is straightforward to show that $0 \le Z_t - Z_{t-1} \le \hat{w}_t$ for all t. The robust counterpart to problem (24) can be formulated as a *linear programming problem:*

$$\begin{aligned} \min \ & \sum_{t=0}^{T-1} (cu_t + y_t) \\ \text{s.t.} \ & y_t \ge h(\bar{x}_{t+1} + Z_t), \quad \forall t \\ & y_t \ge s(-\bar{x}_{t+1} + Z_t), \quad \forall t, \\ & \bar{x}_{t+1} = \bar{x}_t + u_t - \bar{w}_t, \quad \forall t, \\ & u_t \ge 0, \quad \forall t. \end{aligned} \tag{26}$$

A key insight in the analysis of the robust optimization approach is that problem (26) is equivalent to a deterministic inventory problem in which the demand at time t is defined by

$$w'_t = \bar{w}_t + \frac{s-h}{s+h}(Z_t - Z_{t-1}). \tag{27}$$

Therefore, the optimal robust policy is (S, S) with basestock level w'_t. We make the following observations on the robust basestock levels:

- They do not depend on the unit ordering cost, and they depend on the holding and shortage costs only through the ratio $(s-h)/(s+h)$.
- They remain higher, respectively lower, than the nominal ones over the time horizon when shortage is penalized more, respectively less, than holding, and converge towards their nominal values as the time horizon increases.
- They are not constant over time, even when the nominal demands are constant, because they also capture information on the time elapsed since the beginning of the planning horizon.
- They are closer to the nominal basestock values than those obtained in the robust myopic approach (when the robust optimization model only incorporates the next time period); hence, taking into account the whole time horizon mitigates the impact of uncertainty at each time period.

Bertsimas and Thiele [17] provide guidelines to select the budgets of uncertainty based on the worst-case expected cost computed over the set of random demands with given mean and variance. For instance, when $c = 0$ (or $c \ll h, c \ll s$), and the random demands are i.i.d. with mean $\bar{w}$ and standard deviation σ, they take

$$\Gamma_t = \min\left(\frac{\sigma}{\hat{w}}\sqrt{\frac{t+1}{1-\alpha^2}}, t+1\right), \tag{28}$$

with $\alpha = (s-h)/(s+h)$. Equation (28) suggests two phases in the decision-making process:

(1) An early phase in which the decision maker takes a very conservative approach ($\Gamma_t = t+1$),

(2) A later phase in which the decision maker takes advantage of the aggregation of the sources of randomness (Γ_t proportional to $\sqrt{t+1}$).

This is in line with the empirical behavior of the uncertainty observed in Figure 1.

Example 3.1. Inventory Management (Bertsimas and Thiele [17]). For i.i.d. demands with mean 100, standard deviation 20, range forecast $[60, 140]$, a time horizon of 20 periods, and cost parameters $c = 0$, $h = 1$, $s = 3$, the optimal basestock level is given by

$$w_t' = 100 + \frac{20}{\sqrt{3}}(\sqrt{t+1} - \sqrt{t}), \tag{29}$$

which decreases approximately as $1/\sqrt{t}$. Here, the basestock level decreases from 111.5 (for $t = 0$) to 104.8 (for $t = 2$) to 103.7 (for $t = 3$), and ultimately reaches 101.3 ($t = 19$.) The robust optimization framework can incorporate a wide range of additional features, including fixed ordering costs, fixed lead times, integer-order amounts, capacity on the orders, and capacity on the amount in inventory.

3.2.2. Vector Case. We now extend the approach to the case in which the decision maker manages multiple components of the supply chain, such as warehouses and distribution centers. In mathematical terms, the state of the system is described by a vector. While traditional stochastic methods quickly run into tractability issues when the dynamic programming equations are multidimensional, we will see that the robust optimization framework incorporates randomness with no difficulty, in the sense that it can be solved as efficiently as its nominal counterpart. In particular, the robust counterpart of the deterministic inventory management problem remains a linear programming problem, for any topology of the underlying supply network.

We first consider the case in which the system is faced by only one source of uncertainty at each time period, but the state of the system is now described by a vector. A classical example in inventory management arises in *series systems*, where goods proceed through a number of stages (factory, distributor, wholesaler, retailer) before being sold to the customer. We define stage k, $k = 1, \ldots, N$, as the stage in which the goods are k steps away from exiting the network, with stage $k+1$ supplying stage k for $1 \le k \le N-1$. Stage 1 is the stage subject to customer demand uncertainty, and stage N has an infinite supply of goods. Stage k, $k \le N-1$, cannot supply to the next stage more items that it currently has in inventory, which introduces coupling constraints between echelons in the mathematical model. In line with Clark and Scarf [24], we compute the inventory costs at the *echelon* level, with echelon k, $1 \le k \le N$, being defined as the union of all stages from 1 to k as well as the links inbetween. For instance, when the series system represents a manufacturing line where raw materials become work-in-process inventory and ultimately finished products, holding and shortage costs are incurred for items that have reached and possibly moved beyond a given stage in the manufacturing process. Each echelon has the same structure as the single stage described in §3.2.1, with echelon-specific cost parameters.

Bertsimas and Thiele [17] show that

(1) The robust optimization problem can be reformulated as a linear programming problem when there are no fixed ordering costs and a mixed-integer programming problem otherwise.

(2) The optimal policy for echelon k in the robust problem is the same as in a deterministic single-stage problem with modified demand at time t:

$$w_t' = \bar{w}_t + \frac{p_k - h_k}{p_k + h_k}(Z_t - Z_{t-1}), \tag{30}$$

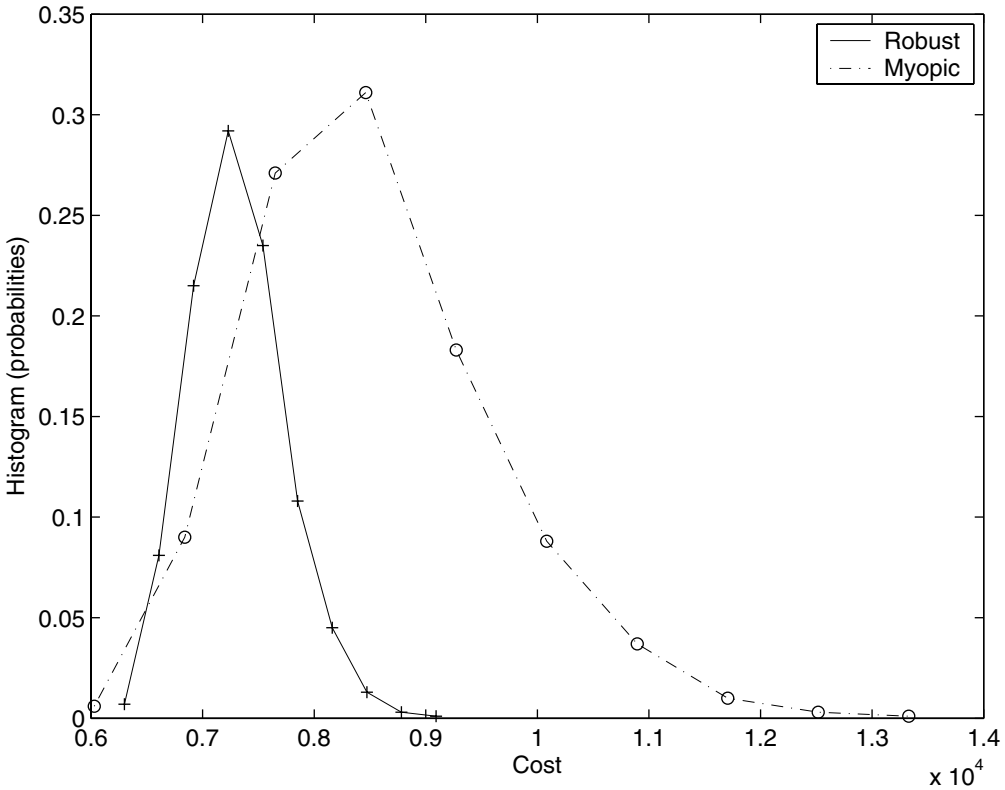
with Z_t defined as in Equation (25), and *time-varying capacity* on the orders.

(3) When there is no fixed ordering cost, the optimal policy for echelon k is the same as in a deterministic *uncapacitated* single-stage problem with demand w'_t at time t and *time-varying cost coefficients*, which depend on the Lagrange multipliers of the coupling constraints. In particular, the policy is basestock.

Hence, the robust optimization approach provides theoretical insights into the impact of uncertainty on the series system, and recovers the optimality of basestock policies established by Clark and Scarf [24] in the stochastic programming framework when there is no fixed ordering costs. It also allows the decision maker to incorporate uncertainty and gain a deeper understanding of problems for which the optimal solution in the stochastic programming framework is *not known*, such as more-complex hierarchical networks. Systems of particular interest are those with an expanding tree structure, because the decision maker can still define echelons in this context and derive some properties on the structure of the optimal solution. Bertsimas and Thiele [17] show that the insights gained for series systems extend to tree networks, where the demand at the retailer is replaced by the cumulative demand at that time period for all retailers in the echelon.

Example 3.2. Inventory Management (Bertsimas and Thiele [17]). A decision maker implements the robust optimization approach on a simple tree network with one warehouse supplying two stores. Ordering costs are all equal to 1, holding and shortage costs at the stores are all equal to 8, while the holding—respectively shortage—costs for the whole system is 5, respectively 7. Demands at the store are i.i.d. with mean 100, standard deviation 20, and range forecast [60, 140]. The stores differ by their initial inventory: 150 and 50 items, respectively, while the whole system initially has 300 items. There are five time periods. Bertsimas and Thiele [17] compare the sample cost of the robust approach with a myopic policy, which adopts a probabilistic description of the randomness at the expense of the time horizon. Figure 4 shows the costs when the myopic policy assumes Gaussian

FIGURE 4. Comparison of costs of robust and myopic policy.



distributions at both stores, which in reality are Gamma with the same mean and variance. Note that the graph for the robust policy is shifted to the left (lower costs) and is narrower than the one for the myopic approach (less volatility).

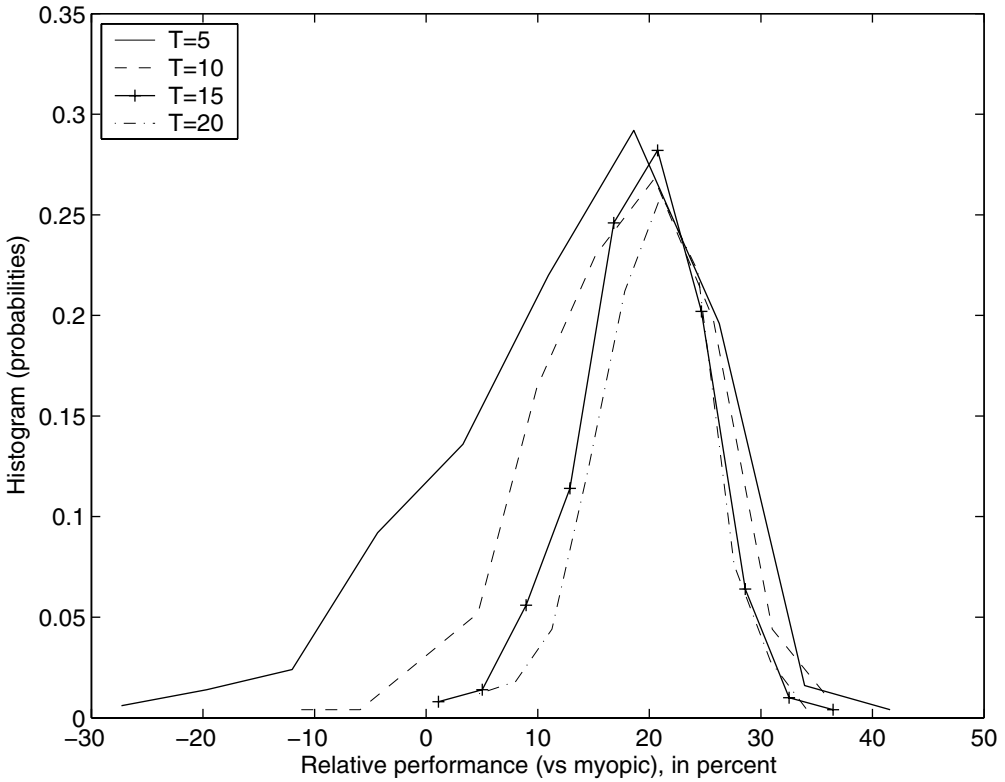
While the error in estimating the distributions to implement the myopic policy is rather small, Figure 4 indicates that not considering the time horizon significantly penalizes the decision maker, even for short horizons as in this example. Figure 5 provides more insights into the impact of the time horizon on the optimal costs. In particular, the distribution of the relative performance between robust and myopic policies shifts to the right of the threshold 0 and becomes narrower (consistently better performance for the robust policy) as the time horizon increases.

These results suggest that taking randomness into account *throughout the time horizon* plays a more important role on system performance than having a detailed probabilistic knowledge of the uncertainty for the next time period.

3.2.3. Dynamic Budgets of Uncertainty. In general, the robust optimization approach we have proposed in §3.2 does not naturally yield policies in dynamic environments and must be implemented on a rolling horizon basis; i.e., the robust problem must be solved repeatedly over time to incorporate new information. In this section, we introduce an extension of this framework proposed by Thiele [46], which (i) allows the decision maker to obtain *policies*, (ii) emphasizes the connection with Bellman's *recursive equations* in stochastic dynamic programming, and (iii) identifies the sources of randomness that affect the system *most negatively*. We present the approach when both state and control variables are scalar and there is only one source of uncertainty at each time period. With similar notation as in §3.2.2, the state variable obeys the linear dynamics given by

$$x_{t+1} = x_t + u_t - w_t, \quad \forall t = 0, \ldots, T-1. \tag{31}$$

FIGURE 5. Impact of the time horizon.



The set of allowable control variables at time t for any state x_t is defined as $U_t(x_t)$. The random variable w_t is modeled as an uncertain parameter with range forecast $[\bar{w}_t - \hat{w}_t, \bar{w}_t + \hat{w}_t]$; the decision maker seeks to protect the system against Γ sources of uncertainty taking their worst-case value over the time horizon. The cost incurred at each time period is the sum of state costs $f_t(x_t)$ and control costs $g_t(u_t)$, where both functions f_t and g_t are convex for all t. Here, we assume that the state costs are computed at the beginning of each time period for simplicity.

The approach hinges on the following question: How should the decision maker spend a budget of uncertainty of Γ units given to him at time 0, and, specifically, for any time period, should he spend one unit of his remaining budget to protect the system against the present uncertainty or keep all of it for future use? To identify the time periods (and states) the decision maker should use his budget on, we consider only three possible values for the uncertain parameter at time t: nominal, highest, and smallest. Equivalently, $w_t = \bar{w}_t + \hat{w}_t z_t$ with $z_t \in \{-1, 0, 1\}$. The *robust counterpart to Bellman's recursive equations* for $t \leq T-1$ is then defined as

$$J_t(x_t, \Gamma_t) = f_t(x_t) + \min_{u_t \in U_t(x_t)} \left[g_t(u_t) + \max_{z_t \in \{-1,\,0,\,1\}} J_t(\bar{x}_{t+1} - \hat{w}_t z_t, \Gamma_t - |z_t|) \right], \quad \Gamma_t \geq 1, \tag{32}$$

$$J_t(x_t, 0) = f_t(x_t) + \min_{u_t \in U_t(x_t)} \left[g_t(u_t) + J_t(\bar{x}_{t+1}, 0) \right]. \tag{33}$$

with the notation $\bar{x}_{t+1} = x_t + u_t - \bar{w}_t$, i.e., $\bar{x}_{t+1}$ is the value taken by the state at the next time period if there is no uncertainty. We also have the boundary equations: $J_T(x_T, \Gamma_T) = f_T(x_T)$ for any x_T and Γ_T. Equations (32) and (33) generate convex problems. Although the cost-to-go functions are now two-dimensional, the approach remains tractable because the cost-to-go function at time t for a budget Γ_t only depends on the cost-to-go function at time $t+1$ for the budgets Γ_t and $\Gamma_t - 1$ (and never for budget values greater than Γ_t). Hence, the recursive equations can be solved by a *greedy algorithm* that computes the cost-to-go functions by increasing the second variable from 0 to Γ and, for each $\gamma \in \{0, \ldots, \Gamma\}$, decreasing the time period from $T-1$ to 0.

Thiele [47] implements this method in revenue management and derives insights into the impact of uncertainty on the optimal policy. Following the same line of thought, Bienstock and Ozbay [21] provide compelling evidence of the tractability of the approach in the context of inventory management.

3.3. Affine and Finite Adaptability

3.3.1. Affine Adaptability. Ben-Tal et al. [10] first extended the robust optimization framework to dynamic settings, where the decision maker adjusts his strategy to information revealed over time using *policies* rather than reoptimization. Their initial focus was on two-stage decision making, which in the stochastic programming literature (e.g., Birge and Louveaux [22]) is referred to as optimization with recourse. Ben-Tal et al. [10] have coined the term "adjustable optimization" for this class of problems when considered in the robust optimization framework. Two-stage problems are characterized by the following sequence of events:

(1) The decision maker selects the "here-and-now," or first-stage, variables, *before* having any knowledge of the actual value taken by the uncertainty;

(2) He observes the realizations of the random variables;

(3) He chooses the "wait-and-see," or second-stage, variables, *after* learning of the outcome of the random event.

In stochastic programming, the sources of randomness obey a discrete, known distribution and the decision maker minimizes the sum of the first-stage and the expected second-stage costs. This is, for instance, justified when the manager can repeat the same experiment

numerous times, has learned the distribution of the uncertainty in the past through historical data, and this distribution does not change. However, such assumptions are rarely satisfied in practice, and the decision maker must then take action with a limited amount of information at his disposal. In that case, an approach based on robust optimization is in order.

The *adjustable robust counterpart* defined by Ben-Tal et al. [10] ensures feasibility of the constraints for any realizations of the uncertainty, through the appropriate selection of the second-stage decision variables $\mathbf{y}(\omega)$, while minimizing (without loss of generality) a deterministic cost:

$$\begin{aligned} \min_{\mathbf{x},\mathbf{y}(\omega)} \ & \mathbf{c}'\mathbf{x} \\ \text{s.t.} \ & \mathbf{A}\mathbf{x} \geq \mathbf{b}, \\ & \mathbf{T}(\omega)\mathbf{x} + \mathbf{W}(\omega)\mathbf{y}(\omega) \geq \mathbf{h}(\omega), \quad \forall\, \omega \in \Omega, \end{aligned} \tag{34}$$

where $\{[\mathbf{T}(\omega), \mathbf{W}(\omega), \mathbf{h}(\omega)], \omega \in \Omega\}$ is a convex uncertainty set describing the possible values taken by the uncertain parameters. In contrast, the *robust counterpart* does not allow for the decision variables to depend on the realization of the uncertainty:

$$\begin{aligned} \min_{\mathbf{x},\mathbf{y}} \ & \mathbf{c}'\mathbf{x} \\ \text{s.t.} \ & \mathbf{A}\mathbf{x} \geq \mathbf{b}, \\ & \mathbf{T}(\omega)\mathbf{x} + \mathbf{W}(\omega)\mathbf{y} \geq \mathbf{h}(\omega), \quad \forall\, \omega \in \Omega. \end{aligned} \tag{35}$$

Ben-Tal et al. [10] show that (i) problems (34) and (35) are equivalent in the case of *constraint-wise uncertainty*, i.e., randomness affects each constraint independently, and (ii) in general, problem (34) is more flexible than problem (35), but this flexibility comes at the expense of tractability (in mathematical terms, problem (34) is *NP-hard*.) To address this issue, the authors propose to restrict the second-stage recourse to be an *affine* function of the realized data; i.e., $\mathbf{y}(\omega) = \mathbf{p} + \mathbf{Q}\omega$ for some $\mathbf{p}, \mathbf{Q}$ to be determined. The *affinely adjustable robust counterpart* is defined as

$$\begin{aligned} \min_{\mathbf{x},\mathbf{p},\mathbf{Q}} \ & \mathbf{c}'\mathbf{x} \\ \text{s.t.} \ & \mathbf{A}\mathbf{x} \geq \mathbf{b}, \\ & \mathbf{T}(\omega)\mathbf{x} + \mathbf{W}(\omega)(\mathbf{p} + \mathbf{Q}\omega) \geq \mathbf{h}(\omega), \quad \forall\, \omega \in \Omega. \end{aligned} \tag{36}$$

In many practical applications, and most of the stochastic programming literature, the recourse matrix $\mathbf{W}(\omega)$ is assumed constant, independent of the uncertainty; this case is known as *fixed recourse*. Using strong duality arguments, Ben-Tal et al. [10] show that problem (36) can be solved efficiently for special structures of the set Ω, in particular, for polyhedra and ellipsoids. In a related work, Ben-Tal et al. [9] implement these techniques for retailer-supplier contracts over a finite horizon and perform a large simulation study, with promising numerical results. Two-stage robust optimization has also received attention in application areas such as network design and operation under demand uncertainty (Atamturk and Zhang [3]).

Affine adaptability has the advantage of providing the decision maker with robust linear *policies*, which are intuitive and relatively easy to implement for well-chosen models of uncertainty. From a theoretical viewpoint, linear decision rules are known to be optimal in *linear-quadratic control*, i.e., control of a system with linear dynamics and quadratic costs (Bertsekas [11]). The main drawback, however, is that there is little justification for the linear decision rule outside this setting. In particular, multistage problems in operations research often yield formulations with *linear* costs and linear dynamics, and because quadratic costs lead to linear (or affine) control, it is not unreasonable when costs are linear to expect good performance from *piecewise constant* decision rules. This claim is motivated by results on the optimal control of fluid models (Ricard [37]).

3.3.2. Finite Adaptability. The concept of finite adaptability, first proposed by Bertsimas and Caramanis [13], is based on the selection of a finite number of (constant) contingency plans to incorporate the information revealed over time. This can be motivated as follows. While robust optimization is well suited for problems where uncertainty is aggregated—i.e., constraintwise—immunizing a problem against uncertainty that cannot be decoupled across constraints yields *overly conservative solutions,* in the sense that the robust approach protects the system against parameters that fall outside the uncertainty set (Soyster [44]). Hence, the decision maker would benefit from gathering some limited information on the actual value taken by the randomness before implementing a strategy. We focus in this tutorial on two-stage models; the framework also has obvious potential in multistage problems.

The recourse under finite adaptability is *piecewise constant* in the number K of contingency plans; therefore, the task of the decision maker is to partition the uncertainty set into K pieces and determine the best response in each. Appealing features of this approach are that (i) it provides a *hierarchy* of adaptability, and (ii) can incorporate integer second-stage variables and nonconvex uncertainty sets, while other proposals of adaptability cannot. We present some of Bertsimas and Caramanis's [13] results below, and in particular, geometric insights into the performance of the K-adaptable approach.

Right-Side Uncertainty. A robust linear programming problem with right-side uncertainty can be formulated as

$$\begin{aligned} \min\ & \mathbf{c}'\mathbf{x} \\ \text{s.t.}\ & \mathbf{A}\mathbf{x} \geq \mathbf{b}, \quad \forall\, \mathbf{b} \in \mathcal{B}, \\ & \mathbf{x} \in \mathcal{X}, \end{aligned} \tag{37}$$

where $\mathcal{B}$ is the polyhedral uncertainty set for the right side of vector $\mathbf{b}$ and $\mathcal{X}$ is a polyhedron, not subject to uncertainty. To ensure that the constraints $\mathbf{A}\mathbf{x} \geq \mathbf{b}$ hold for all $\mathbf{b} \in \mathcal{B}$, the decision maker must immunize each constraint i against uncertainty:

$$\mathbf{a}_i'\mathbf{x} \geq b_i, \quad \forall\, \mathbf{b} \in \mathcal{B}, \tag{38}$$

which yields

$$\mathbf{A}\mathbf{x} \geq \tilde{\mathbf{b}}_0, \tag{39}$$

where $(\tilde{b}_0)_i = \max\{b_i \mid \mathbf{b} \in \mathcal{B}\}$ for all i. Therefore, solving the robust problem is equivalent to solving the deterministic problem with the right side being equal to $\tilde{\mathbf{b}}_0$. Note that $\tilde{\mathbf{b}}_0$ is the "upper-right" corner of the smallest hypercube $\mathcal{B}_0$ containing $\mathcal{B}$, and might fall far outside the uncertainty set. In that case, nonadjustable robust optimization forces the decision maker to plan for a very unlikely outcome, which is an obvious drawback to the adoption of the approach by practitioners.

To address the issue of overconservatism, Bertsimas and Caramanis [13] cover the uncertainty set $\mathcal{B}$ with a partition of K (not necessarily disjoint) pieces: $\mathcal{B} = \bigcup_{k=1}^{K} \mathcal{B}_k$, and select a contingency plan $\mathbf{x}_k$ for each subset $\mathcal{B}_k$. The *K-adaptable robust counterpart* is defined as

$$\begin{aligned} \min\ & \max_{k=1,\ldots,K} \mathbf{c}'\mathbf{x}_k \\ \text{s.t.}\ & \mathbf{A}\mathbf{x}_k \geq \mathbf{b}, \quad \forall\, \mathbf{b} \in \mathcal{B}_k, \quad \forall\, k = 1, \ldots, K, \\ & \mathbf{x}_k \in \mathcal{X}, \quad \forall\, k = 1, \ldots, K. \end{aligned} \tag{40}$$

It is straightforward to see that problem (40) is equivalent to

$$\begin{aligned} \min\ & \max_{k=1,\ldots,K} \mathbf{c}'\mathbf{x}_k \\ \text{s.t.}\ & \mathbf{A}\mathbf{x}_k \geq \tilde{\mathbf{b}}_k, \quad \forall\, k = 1, \ldots, K, \\ & \mathbf{x}_k \in \mathcal{X}, \quad \forall\, k = 1, \ldots, K, \end{aligned} \tag{41}$$

where $\tilde{\mathbf{b}}_k$ is defined as $(\tilde{b}_k)_i = \max\{b_i \mid \mathbf{b} \in \mathcal{B}_k\}$ for each i, and represents the upper-right corner of the smallest hypercube containing $\mathcal{B}_k$. Hence, the performance of the finite adaptability approach depends on the choice of the subsets $\mathcal{B}_k$ only through the resulting value of $\tilde{\mathbf{b}}_k$, with $k = 1, \ldots, K$. This motivates developing a direct connection between the uncertainty set $\mathcal{B}$ and the vectors $\tilde{\mathbf{b}}_k$, without using the subsets $\mathcal{B}_k$.

Let $\mathcal{C}(\mathcal{B})$ be the set of K-uples $(\mathbf{b}_1, \ldots, \mathbf{b}_K)$ covering the set $\mathcal{B}$, i.e., for any $\mathbf{b} \in \mathcal{B}$,— the inequality $\mathbf{b} \leq \mathbf{b}_k$ holds for at least one k. The problem of optimally partitioning the uncertainty set into K pieces can be formulated as

$$\begin{aligned} \min \ & \max_{k=1,\ldots,K} \mathbf{c}'\mathbf{x}_k \\ \text{s.t.} \ & \mathbf{A}\mathbf{x}_k \geq \tilde{\mathbf{b}}_k, \quad \forall k = 1, \ldots, K, \\ & \mathbf{x}_k \in \mathcal{X}, \quad \forall k = 1, \ldots, K, \\ & (\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_K) \in \mathcal{C}(\mathcal{B}). \end{aligned} \tag{42}$$

The characterization of $\mathcal{C}(\mathcal{B})$ plays a central role in the approach. Bertsimas and Caramanis [13] investigate in detail the case with two contingency plans, where the decision maker must select a pair $(\tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_2)$ that covers the set $\mathcal{B}$. For any $\tilde{\mathbf{b}}_1$, the vector $\min(\tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_0)$ is also feasible and yields a smaller or equal cost in problem (42). A similar argument holds for $\tilde{\mathbf{b}}_2$. Hence, the optimal $(\tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_2)$ pair in Equation (42) satisfies $\tilde{\mathbf{b}}_1 \leq \tilde{\mathbf{b}}_0$ and $\tilde{\mathbf{b}}_2 \leq \tilde{\mathbf{b}}_0$. On the other hand, for $(\tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_2)$ to cover $\mathcal{B}$, we must have either $b_i \leq \tilde{b}_{1i}$ or $b_i \leq \tilde{b}_{2i}$ for each component i of any $\mathbf{b} \in \mathcal{B}$. Hence, for each i, either $\tilde{b}_{1i} = \tilde{b}_{0i}$ or $\tilde{b}_{2i} = \hat{b}_{0i}$.

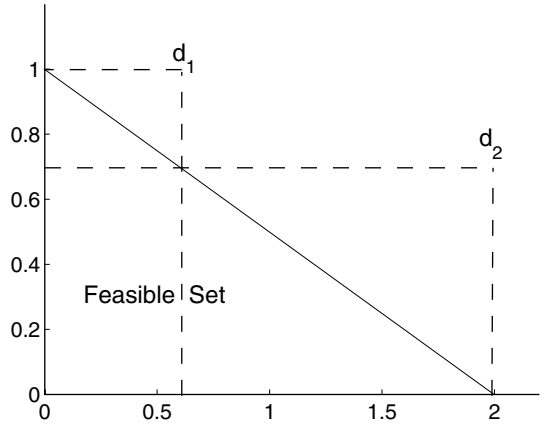
This creates a partition S between the indices $\{1, \ldots, n\}$, where $S = \{i \mid \tilde{b}_{1i} = \tilde{b}_{0i}\}$. $\tilde{\mathbf{b}}_1$ is completely characterized by the set S, in the sense that $\tilde{b}_{1i} = \tilde{b}_{0i}$ for all $i \in S$ and $\tilde{b}_{1i}$ for $i \notin S$ can be any number smaller than $\tilde{b}_{0i}$. The part of $\mathcal{B}$ that is not yet covered is $\mathcal{B} \cap \{\exists j, b_j \geq \tilde{b}_{1j}\}$. This forces $\tilde{b}_{2i} = \tilde{b}_{0i}$ for all $i \notin S$ and $\tilde{b}_{2i} \geq \max\{b_i \mid \mathbf{b} \in \mathcal{B}, \exists j \in S^c, b_j \geq \tilde{b}_{1j}\}$, or equivalently, $\tilde{b}_{2i} \geq \max_j \max\{b_i \mid \mathbf{b} \in \mathcal{B}, b_j \geq \tilde{b}_{1j}\}$, for all $i \in S$. Bertsimas and Caramanis [13] show that

- When $\mathcal{B}$ has a specific structure, the optimal split and corresponding contingency plans can be computed as the solution of a mixed integer-linear program.
- Computing the optimal partition is NP-hard, but can be performed in a tractable manner when either of the following quantities is small: the dimension of the uncertainty, the dimension of the problem, or the number of constraints affected by the uncertainty.
- When none of the quantities above is small, a well-chosen heuristic algorithm exhibits strong empirical performance in large-scale applications.

Example 3.3. Newsvendor Problem with Reorder. A manager must order two types of seasonal items before knowing the actual demand for these products. All demand must be met; therefore, once demand is realized, the missing items (if any) are ordered at a more-expensive reorder cost. The decision maker considers two contingency plans. Let x_j, $j = 1, 2$ be the amounts of product j ordered before demand is known, and y_{ij} the amount of product j ordered in contingency plan i, $i = 1, 2$. We assume that the first-stage ordering costs are equal to 1 and the second-stage ordering costs are equal to 2. Moreover, the uncertainty set for the demand is given by $\{(d_1, d_2) \mid d_1 \geq 0, d_2 \geq 0, d_1/2 + d_2 \leq 1\}$.

The robust, static counterpart would protect the system against $d_1 = 2$, $d_2 = 1$, which falls outside the feasible set, and would yield an optimal cost of 3. To implement the two-adaptability approach, the decision maker must select an optimal covering pair $(\tilde{\mathbf{d}}_1, \tilde{\mathbf{d}}_2)$ satisfying $\tilde{\mathbf{d}}_1 = (d, 1)$ with $0 \leq d \leq 2$ and $\tilde{\mathbf{d}}_2 = (1, d')$ with $d' \geq 1 - d/2$. At optimality, $d' = 1 - d/2$, because increasing the value of d' above that threshold increases the optimal cost while the demand uncertainty set is already completely covered. Hence, the partition is determined by the scalar d. Figure 6 depicts the uncertainty set and a possible partition.

FIGURE 6. The uncertainty set and a possible partition.



The two-adaptable problem can be formulated as

$$\begin{aligned}
\min\ & Z \\
\text{s.t.}\ & Z \geq x_1 + x_2 + 2(y_{11} + y_{12}), \\
& Z \geq x_1 + x_2 + 2(y_{21} + y_{22}), \\
& x_1 + y_{11} \geq d, \\
& x_2 + y_{12} \geq 1, \\
& x_1 + y_{21} \geq 1, \\
& x_2 + y_{22} \geq 1 - d/2, \\
& x_j, y_{ij} \geq 0, \quad \forall i, j, \\
& 0 \leq d \leq 2.
\end{aligned} \tag{43}$$

The optimal solution is to select $d = 2/3$, $\mathbf{x} = (2/3, 2/3)$ and $\mathbf{y}_1 = (0, 1/3)$, $\mathbf{y}_2 = (1/3, 0)$, for an optimal cost of 2. Hence, two-adaptability achieves a decrease in cost of 33%.

Matrix Uncertainty. In this paragraph, we briefly outline Bertsimas and Caramanis's [13] findings in the case of matrix uncertainty and two-adaptability. For notational convenience, we incorporate constraints without uncertainty ($\mathbf{x} \in \mathcal{X}$ for a given polyhedron $\mathcal{X}$) into the constraints $\mathbf{Ax} \geq \mathbf{b}$. The robust problem can be written as

$$\begin{aligned}
\min\ & \mathbf{c}'\mathbf{x} \\
\text{s.t.}\ & \mathbf{Ax} \geq \mathbf{b}, \quad \forall\, \mathbf{A} \in \mathcal{A},
\end{aligned} \tag{44}$$

where the uncertainty set $\mathcal{A}$ is a polyhedron. Here, we define $\mathcal{A}$ by its extreme points: $\mathcal{A} = conv\{\mathbf{A}_1, \ldots, \mathbf{A}_K\}$, where *conv* denotes the convex hull. Problem (44) becomes

$$\begin{aligned}
\min\ & \mathbf{c}'\mathbf{x} \\
\text{s.t.}\ & \mathbf{A}_k\mathbf{x} \geq \mathbf{b}, \quad \forall\, k = 1, \ldots, K.
\end{aligned} \tag{45}$$

Let $\mathcal{A}_0$ be the smallest hypercube containing $\mathcal{A}$. We formulate the two-adaptability problem as

$$\begin{aligned}
\min\ & \max\{\mathbf{c}'\mathbf{x}_1, \mathbf{c}'\mathbf{x}_2\} \\
\text{s.t.}\ & \mathbf{Ax}_1 \geq \mathbf{b}, \quad \forall\, \mathbf{A} \in \mathcal{A}_1, \\
& \mathbf{Ax}_2 \geq \mathbf{b}, \quad \forall\, \mathbf{A} \in \mathcal{A}_2,
\end{aligned} \tag{46}$$

where $\mathcal{A} \subset (\mathcal{A}_1 \cup \mathcal{A}_2) \subset \mathcal{A}_0$.

Bertsimas and Caramanis [13] investigate in detail the conditions for which the two-adaptable approach improves the cost of the robust static solution by at least $\eta > 0$. Let $\mathbf{A}_0$ be the corner point of $\mathcal{A}_0$ such that problem (44) is equivalent to $\min\{\mathbf{c}'\mathbf{x} \text{ s.t. } \mathbf{A}_0\mathbf{x} \geq \mathbf{b}\}$. Intuitively, the decision maker needs to remove from the partition $\mathcal{A}_1 \cup \mathcal{A}_2$ an area around $\mathbf{A}_0$ large enough to ensure this cost decrease. The authors build on this insight to provide a geometric perspective on the gap between the robust and the two-adaptable frameworks. A key insight is that, if v^* is the optimal objective of the robust problem (44), the problem

$$\begin{aligned} \min\ & 0 \\ \text{s.t}\ & \mathbf{A}^i\mathbf{x} \geq \mathbf{b}, \quad \forall\, i = 1, \ldots, K, \\ & \mathbf{c}'\mathbf{x} \leq v^* - \eta \end{aligned} \tag{47}$$

is infeasible. Its dual is feasible (for instance, $\mathbf{0}$ belongs to the feasible set) and hence unbounded by strong duality. The set $\mathcal{D}$ of directions of dual unboundedness is obtained by scaling the extreme rays:

$$\mathcal{D} = \left\{ (\mathbf{p}_1, \ldots, \mathbf{p}_K) \;\middle|\; \mathbf{b}'\left(\sum_{i=1}^{K} \mathbf{p}_i\right) \geq v^* - \eta, \sum_{i=1}^{K} (\mathbf{A}^i)'\mathbf{p}_i = \mathbf{c}, \mathbf{p}_1, \ldots, \mathbf{p}_K \geq 0. \right\}. \tag{48}$$

The $(\mathbf{p}_1, \ldots, \mathbf{p}_K)$ in the set $\mathcal{D}$ are used to construct a family $\mathcal{A}_\eta$ of matrices $\widetilde{\mathbf{A}}$ such that the optimal cost of the nominal problem (solved for any matrix in this family) is at least equal to $v^* - \eta$. (This is simply done by defining $\widetilde{\mathbf{A}}$ such that $\sum_{i=1}^{K} \mathbf{p}_i$ is feasible for the dual of the nominal problem, i.e., $\widetilde{\mathbf{A}}' \sum_{i=1}^{K} \mathbf{p}_i = \sum_{i=1}^{K} (\mathbf{A}^i)'\mathbf{p}_i$.) The family $\mathcal{A}_\eta$ plays a crucial role in understanding the performance of the two-adaptable approach. Specifically, two-adaptability decreases the cost by strictly more than η if and only if $\mathcal{A}_\eta$ has no element in the partition $\mathcal{A}_1 \cup \mathcal{A}_2$. The reader is referred to Bertsimas and Caramanis [13] for additional properties.

As pointed out in Bertsimas and Caramanis [13], finite adaptability is *complementary* to the concept of affinely adjustable optimization proposed by Ben-Tal et al. [10], in the sense that neither technique performs consistently better than the other. Understanding the problem structure required for good performance of these techniques is an important future research direction. Bertsimas et al. [19] apply the adaptable framework to air traffic control subject to weather uncertainty, where they demonstrate the method's ability to incorporate randomness in very large-scale integer formulations.

4. Connection with Risk Preferences

4.1. Robust Optimization and Coherent Risk Measures

So far, we have assumed that the polyhedral set describing the uncertainty was *given*, and developed robust optimization models based on that input. In practice, however, the true information available to the decision maker is historical data, which must be incorporated into an uncertainty set before the robust optimization approach can be implemented. We now present an explicit methodology to construct this set, based on past observations of the random variables and the decision maker's attitude toward risk. The approach is due to Bertsimas and Brown [12]. An application of data-driven optimization to inventory management is presented in Bertsimas and Thiele [16].

We consider the following problem:

$$\begin{aligned} \min\ & \mathbf{c}'\mathbf{x} \\ \text{s.t.}\ & \mathbf{a}'\mathbf{x} \leq b, \\ & \mathbf{x} \in \mathcal{X}. \end{aligned} \tag{49}$$

The decision maker has N historical observations $\mathbf{a}_1, \ldots, \mathbf{a}_N$ of the random vector $\tilde{\mathbf{a}}$ at his disposal. Therefore, for any given $\mathbf{x}$, $\tilde{\mathbf{a}}'\mathbf{x}$ is a random variable whose sample distribution is given by $P[\tilde{\mathbf{a}}'\mathbf{x} = \mathbf{a}_i'\mathbf{x}] = 1/N$, for $i = 1, \ldots, N$. (We assume that the $\mathbf{a}_i'\mathbf{x}$ are distinct, and the extension to the general case is straightforward.) The decision maker associates a numerical value $\mu(\tilde{\mathbf{a}}'\mathbf{x})$ to the random variable $\tilde{\mathbf{a}}'\mathbf{x}$; the function μ captures his attitude toward risk and is called a *risk measure*. We then define the *risk-averse problem* as

$$\begin{aligned} \min\ & \mathbf{c}'\mathbf{x} \\ \text{s.t.}\ & \mu(\tilde{\mathbf{a}}'\mathbf{x}) \le b, \\ & \mathbf{x} \in \mathcal{X}. \end{aligned} \tag{50}$$

While any function from the space of almost surely bounded random variables $\mathcal{S}$ to the space of real numbers $\mathcal{R}$ can be selected as a risk measure, some are more sensible choices than others. In particular, Artzner et al. [1] argue that a measure of risk should satisfy four axioms, which define the class of *coherent risk measures*:

(1) Translation invariance: $\mu(X + a) = \mu(X) - a, \forall\, X \in \mathcal{S}, a \in \mathcal{R}$.
(2) Monotonicity: if $X \le Y$ w.p. 1, $\mu(X) \le \mu(Y), \forall\, X, Y \in \mathcal{S}$.
(3) Subadditivity: $\mu(X + Y) \le \mu(X) + \mu(Y), \forall\, X, Y \in \mathcal{S}$.
(4) Positive homogeneity: $\mu(\lambda X) = \lambda\mu(X), \forall\, X \in \mathcal{S}, \lambda \ge 0$.

An example of a coherent risk measure is the tail conditional expectation, i.e., the expected value of the losses given that they exceed some quantile. Other risk measures such as standard deviation and the probability that losses will exceed a threshold, also known as value-at-risk, are not coherent for general probability distributions.

An important property of coherent risk measures is that they can be represented as the *worst-case expected value over a family of distributions.* Specifically, μ is coherent if and only if there exists a family of probability measures $\mathcal{Q}$ such that

$$\mu(X) = \sup_{q \in \mathcal{Q}} E_q[X], \quad \forall\, X \in \mathcal{S}. \tag{51}$$

In particular, if μ is a coherent risk measure and $\tilde{\mathbf{a}}$ is distributed according to its sample distribution ($P(\mathbf{a} = \mathbf{a}_i) = 1/N$ for all i), Bertsimas and Brown [12] note that

$$\mu(\tilde{\mathbf{a}}'\mathbf{x}) = \sup_{q \in \mathcal{Q}} E_Q[\tilde{\mathbf{a}}'\mathbf{x}] = \sup_{q \in \mathcal{Q}} \sum_{i=1}^{N} q_i \mathbf{a}_i'\mathbf{x} = \sup_{a \in \mathcal{A}} \mathbf{a}'\mathbf{x}, \tag{52}$$

with the uncertainty set $\mathcal{A}$ defined by

$$\mathcal{A} = conv\left\{ \sum_{i=1}^{N} q_i \mathbf{a}_i \;\middle|\; \mathbf{q} \in \mathcal{Q} \right\}, \tag{53}$$

and the risk-averse problem (50) is then *equivalent* to the robust optimization problem:

$$\begin{aligned} \min\ & \mathbf{c}'\mathbf{x} \\ \text{s.t.}\ & \mathbf{a}'\mathbf{x} \le b, \quad \forall\, \mathbf{a} \in \mathcal{A}, \\ & \mathbf{x} \in \mathcal{X}. \end{aligned} \tag{54}$$

The convex (not necessarily polyhedral) uncertainty set $\mathcal{A}$ is included into the convex hull of the data points $\mathbf{a}_1, \ldots, \mathbf{a}_N$. Equation (53) provides an *explicit characterization* of the uncertainty set that the decision maker should use if her/his attitude toward risk is based on a coherent risk measure. It also raises two questions: (i) Can we obtain the generating family $\mathcal{Q}$ easily, at least for some well-chosen coherent risk measures? (ii) Can we identify risk measures that lead to polyhedral uncertainty sets, because those sets have been central to the robust optimization approach presented so far? In §4.2, we address both issues simultaneously by introducing the concept of *comonotone risk measures.*

4.2. Comonotone Risk Measures

To investigate the connection between the decision maker's attitude toward risk and the choice of polyhedral uncertainty sets, Bertsimas and Brown [12] consider a second representation of coherent risk measures based on *Choquet integrals.*

The Choquet integral μ_g of a random variable $X \in \mathcal{S}$ with respect to the distortion function g (which can be any nondecreasing function on $[0,1]$ such that $g(0)=0$ and $g(1)=1$) is defined by

$$\mu_g(X) = \int_0^\infty g(P[X \geq x])\,dx + \int_{-\infty}^0 [g(P[X \geq x]) - 1]\,dx. \tag{55}$$

μ_g is coherent if and only if g is concave (Reesor and McLeish [36]). While not every coherent risk measure can be recast as the expected value of a random variable under a distortion function, Choquet integrals provide a broad modeling framework, which includes conditional tail expectation and value-at-risk. Schmeidler [39] shows that a risk measure can be represented as a Choquet integral with a concave distortion function (and hence be coherent) if and only if the risk measure satisfies a property called *comonotonicity.*

A random variable is said to be comonotonic if its support S has a complete order structure (for any $\mathbf{x}, \mathbf{y} \in S$, either $\mathbf{x} \leq \mathbf{y}$ or $\mathbf{y} \leq \mathbf{x}$), and a risk measure is said to be comonotone if for any comonotonic random variables X and Y, we have

$$\mu(X+Y) = \mu(X) + \mu(Y). \tag{56}$$

Example 4.1. Comonotonic Random Variable (Bertsimas and Brown [12]). Consider the joint payoff of a stock and a call option on that stock. With S the stock value and K the strike price of the call option, the joint payoff $(S, \max(0, S-K))$ is obviously comonotonic. For instance, with $K=2$ and S taking any value between 1 and 5, the joint payoff takes values $\mathbf{x}_1=(1,0)$, $\mathbf{x}_2=(2,0)$, $\mathbf{x}_3=(3,1)$, $\mathbf{x}_4=(4,2)$, and $\mathbf{x}_5=(5,3)$. Hence, $\mathbf{x}_{i+1} \geq \mathbf{x}_i$ for each i.

Bertsimas and Brown [12] show that, for any comonotone risk measure with distortion function g, noted μ_g, and any random variable Y with support $\{y_1, \ldots, y_N\}$ such that $P[Y=y_i]=1/N$, μ_g can be computed using the formula

$$\mu_g(Y) = \sum_{i=1}^N q_i y_{(i)}, \tag{57}$$

where $y_{(i)}$ is the ith smallest y_j, $j=1,\ldots,N$ (hence, $y_{(1)} \leq \cdots \leq y_{(N)}$), and q_i is defined by

$$q_i = g\left(\frac{N+1-i}{N}\right) - g\left(\frac{N-i}{N}\right). \tag{58}$$

Because g is nondecreasing and concave, it is easy to see that the q_i are nondecreasing. Bertsimas and Brown [12] use this insight to represent $\sum_{i=1}^N q_i y_{(i)}$ as the optimal solution of a linear programming problem

$$\begin{aligned}
\max \ & \sum_{i=1}^N \sum_{j=1}^N q_i y_j w_{ij} \\
\text{s.t.} \ & \sum_{i=1}^N w_{ij} = 1, \quad \forall j, \\
& \sum_{j=1}^N w_{ij} = 1, \quad \forall i, \\
& w_{ij} \geq 0, \quad \forall i,j.
\end{aligned} \tag{59}$$

At optimality, the largest y_i is assigned to q_N, the second largest to q_{N-1}, and so on. Let $W(N)$ be the feasible set of problem (59). Equation (57) becomes

$$\mu_g(Y) = \max_{\mathbf{w} \in W(N)} \sum_{i=1}^{N} \sum_{j=1}^{N} q_i y_j w_{ij}. \tag{60}$$

This yields a generating family $\mathcal{Q}$ for μ_g:

$$\mathcal{Q} = \{\mathbf{w}'\mathbf{q}, \mathbf{w} \in W(N)\}, \tag{61}$$

or equivalently, using the optimal value of $\mathbf{w}$:

$$\mathcal{Q} = \{\mathbf{p}, \exists \sigma \in S_N, p_i = q_{\sigma(i)}, \forall i\}, \tag{62}$$

where S_N is the group of permutations over $\{1, \ldots, N\}$. Bertsimas and Brown [12] make the following observations:

- While coherent risk measures are in general defined by a *family* $\mathcal{Q}$ of probability distributions, comonotone risk measures require the knowledge of a *single generating vector* $\mathbf{q}$. The family $\mathcal{Q}$ is then derived according to Equation (62).
- Comonotone risk measures lead to polyhedral uncertainty sets of a *specific structure*: the convex hull of all $N!$ convex combinations of $\{\mathbf{a_1}, \ldots, \mathbf{a_N}\}$ induced by all permutations of the vector $\mathbf{q}$.

It follows from injecting the generating family $\mathcal{Q}$ given by Equation (62) into the definition of the uncertainty set $\mathcal{A}$ in Equation (53) that the risk-averse problem (50) is equivalent to the robust optimization problem solved for the polyhedral uncertainty set:

$$\mathcal{A}_q = conv\left\{\sum_{i=1}^{N} q_{\sigma(i)} a_i, \sigma \in S_N\right\}. \tag{63}$$

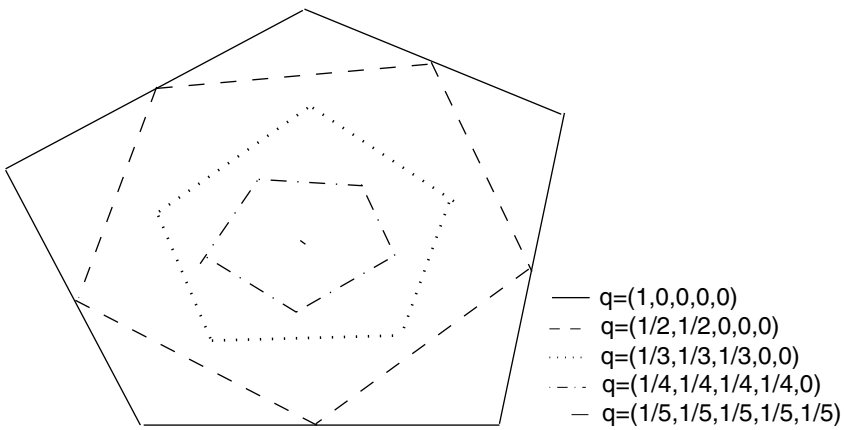
Note that $\mathbf{q} = (1/N)\mathbf{e}$ with $\mathbf{e}$ the vector of all one's yields the sample average $(1/N)\sum_{i=1}^{N} \mathbf{a}_i$ and $\mathbf{q} = (1, 0, \ldots, 0)$ yields the convex hull of the data. Figure 7 shows possible uncertainty sets with $N = 5$ observations.

4.3. Additional Results

Bertsimas and Brown [12] provide a number of additional results connecting coherent risk measures and convex uncertainty sets. We enumerate a few here:

(1) Tail conditional expectations $CTE_{i/N}$, $i = 1, \ldots, N$, can be interpreted as *basis functions* for the entire space of comonotone risk measures on random variables with a discrete state space of size N.

FIGURE 7. Uncertainty sets derived from comonotone risk measures.



(2) The class of *symmetric polyhedral uncertainty sets* is generated by a specific set of coherent risk measures. These uncertainty sets are useful because they naturally induce a norm.

(3) Optimization over the following coherent risk measure based on higher-order tail moments:

$$\mu_{p,\alpha}(X) = E[X] + \alpha(E[(\max\{0, X - E[X]\})^p])^{1/p} \tag{64}$$

is equivalent to a robust optimization problem with a norm-bounded uncertainty set.

(4) Any robust optimization problem with a convex uncertainty set (contained within the convex hull of the data) can be reformulated as a risk-averse problem with a coherent risk measure.

5. Conclusions

Robust optimization has emerged over the last decade as a tractable, insightful approach to decision making under uncertainty. It is well-suited for both static and dynamic problems with imprecise information; has a strong connection with the decision maker's attitude toward risk, and can be applied in numerous areas, including inventory management, air traffic control, revenue management, network design, and portfolio optimization. While this tutorial has primarily focused on linear programming and polyhedral uncertainty sets, the modeling power of robust optimization extends to more general settings, for instance, second-order cone programming and ellipsoidal uncertainty sets. It has also been successfully implemented in stochastic and dynamic programming with ambiguous probabilities. Current topics of interest include (i) tractable methods to incorporate information revealed over time in multistage problems, and (ii) data-driven optimization, which injects historical data directly into the mathematical programming model—for instance, through explicit guidelines to construct the uncertainty set. Hence, the robust and data-driven framework provides a compelling alternative to traditional decision-making techniques under uncertainty.

References

[1] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath. Coherent measures of risk. *Mathematical Finance* 9(3):203–228, 1999.

[2] A. Atamturk. Strong formulations of robust mixed 0-1 programming. *Mathematical Programming* 108(2–3):235–250, 2005.

[3] A. Atamturk and M. Zhang. Two-stage robust network flow and design under demand uncertainty. Technical report, University of California, Berkeley, CA, 2004.

[4] A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research* 23(4):769–805, 1998.

[5] A. Ben-Tal and A. Nemirovski. Robust solutions to uncertain programs. *Operations Research Letters* 25:1–13, 1999.

[6] A. Ben-Tal and A. Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming* 88:411–424, 2000.

[7] A. Ben-Tal, S. Boyd, and A. Nemirovski. Extending the scope of robust optimization: Comprehensive robust counterparts of uncertain problems. Technical report, Georgia Institute of Technology, Atlanta, GA, 2005.

[8] A. Ben-Tal, A. Nemirovski, and C. Roos. Robust solutions of uncertain quadratic and conic-quadratic problems. *SIAM Journal on Optimization* 13(535–560), 2002.

[9] A. Ben-Tal, B. Golani, A. Nemirovski, and J.-P. Vial. Supplier-retailer flexible commitments contracts: A robust optimization approach. *Manufacturing and Service Operations Management* 7(3):248–273, 2005.

[10] A. Ben-Tal, A. Goryashko, E. Guslitser, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming* 99:351–376, 2004.

[11] D. Bertsekas. *Dynamic Programming and Optimal Control*, Vol. 1, 2nd ed. Athena Scientific, Belmont, MA, 2001.

[12] D. Bertsimas and D. Brown. Robust linear optimization and coherent risk measures. Technical report, Massachusetts Institute of Technology, Cambridge, MA, 2005.

[13] D. Bertsimas and C. Caramanis. Finite adaptability in linear optimization. Technical report, Massachusetts Institute of Technology, Cambridge, MA, 2005.

[14] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming* 98:49–71, 2003.

[15] D. Bertsimas and M. Sim. The price of robustness. *Operations Research* 52(1):35–53, 2004.

[16] D. Bertsimas and A. Thiele. A data-driven approach to newsvendor problems. Technical report, Massachusetts Institute of Technology, Cambridge, MA, 2004.

[17] D. Bertsimas and A. Thiele. A robust optimization approach to inventory theory. *Operations Research* 54(1):150–168, 2006.

[18] D. Bertsekas and J. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA, 1996.

[19] D. Bertsimas, C. Caramanis, and W. Moser. Multistage finite adaptability: Application to air traffic control. Working paper, Massachusetts Institute of Technology, Cambridge, MA, 2006.

[20] D. Bertsimas, D. Pachamanova, and M. Sim. Robust linear optimization under general norms. *Operations Research Letters* 32(6):510–516, 2004.

[21] D. Bienstock and N. Ozbay. Computing optimal basestocks. Technical report, Columbia University, New York, 2005.

[22] J. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer Verlag, New York, 1997.

[23] A. Charnes and W. Cooper. Chance-constrained programming. *Management Science* 6(1): 73–79, 1959.

[24] A. Clark and H. Scarf. Optimal policies for a multi-echelon inventory problem. *Management Science* 6(4):475–490, 1960.

[25] G. Dantzig. Linear programming under uncertainty. *Management Science* 1(3–4):197–206, 1955.

[26] J. Dupačová. The minimax approach to stochastic programming and an illustrative application. *Stochastics* 20:73–88, 1987.

[27] L. El-Ghaoui and H. Lebret. Robust solutions to least-square problems to uncertain data matrices. *SIAM Journal on Matrix Analysis and Applications* 18:1035–1064, 1997.

[28] L. El-Ghaoui, F. Oustry, and H. Lebret. Robust solutions to uncertain semidefinite programs. *SIAM Journal on Optimization* 9:33–52, 1998.

[29] D. Goldfarb and G. Iyengar. Robust portfolio selection problems. *Mathematics of Operations Research* 28(1):1–38, 2003.

[30] G. Iyengar. Robust dynamic programming. *Mathematics of Operations Research* 30(2):257–280, 2005.

[31] P. Kall and J. Mayer. *Stochastic Linear Programming: Models, Theory and Computation*. Springer-Verlag, New York, 2005.

[32] S. Nahmias. *Production and Operations Analysis*, 5th ed. McGraw-Hill, New York, 2005.

[33] A. Nilim and L. El-Ghaoui. Robust control of Markov decision processes with uncertain transition matrices. *Operations Research* 53(5):780–798, 2005.

[34] F. Ordonez and J. Zhao. Robust capacity expansion of network flows. Technical report, University of Southern California, Los Angeles, CA, 2005.

[35] E. Porteus. *Foundations of Stochastic Inventory Theory*. Stanford University Press, Palo Alto, CA, 2002.

[36] M. Reesor and D. McLeish. Risk, entropy and the transformation of distributions. Technical report, Bank of Canada, Ottawa, Ontario, Canada, 2002.

[37] M. Ricard. *Optimization of Queueing Networks, an Optimal Control Approach*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 1995.

[38] H. Scarf. A min-max solution of an inventory problem. *Studies in the Mathematical Theory of Inventory and Production*. Stanford University Press, Stanford, CA, 201–209, 1958.

[39] D. Schmeidler. Integral representation without additivity. *Proceedings of the American Mathematical Society*, 97:255–261, 1986.

[40] A. Shapiro. Worst-case distribution analysis of stochastic programs. *Mathematical Programming*, 107(1–2):91–96, 2006.

[41] Y. Sheffi. *The Resilient Enterprise: Overcoming Vulnerability for Competitive Advantage*. MIT Press, Cambridge, MA, 2005.

[42] M. Sim. Robust optimization. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 2004.

[43] D. Simchi-Levi, P. Kaminsky, and E. Simchi-Levi. *Managing the Supply Chain: The Definitive Guide for the Business Professional*. McGraw-Hill, New York, 2004.

[44] A. Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research* 21:1154–1157, 1973.

[45] A. Thiele. A robust optimization approach to supply chain and revenue management. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 2004.

[46] A. Thiele. Robust dynamic optimization: A distribution-free approach. Technical report, Lehigh University, Bethlehem, PA, 2005.

[47] A. Thiele. Robust revenue management with dynamic budgets of uncertainty. Technical report, Lehigh University, Bethlehem, PA, 2005.

[48] J. Žáčková. On minimax solutions of stochastic linear programming problems. *Časopis pro Pěstování Matematiky* 91:423–430, 1966.

TUTORIALS IN
OPERATIONS RESEARCH
INFORMS 2006

 | ISBN 13 978-1-877640-20-9
DOI 10.1287/educ.1063.0027

Approximate Dynamic Programming for Large-Scale Resource Allocation Problems

Warren B. Powell
Department of Operations Research and Financial Engineering, Princeton University, Princeton, New Jersey 08544, powell@princeton.edu

Huseyin Topaloglu
School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York 14853, topaloglu@orie.cornell.edu

Abstract We present modeling and solution strategies for large-scale resource allocation problems that take place over multiple time periods under uncertainty. In general, the strategies we present formulate the problem as a dynamic program and replace the value functions with tractable approximations. The approximations of the value functions are obtained by using simulated trajectories of the system and iteratively improving on (possibly naive) initial approximations; we propose several improvement algorithms for this purpose. As a result, the resource allocation problem decomposes into time-staged subproblems, where the impact of the current decisions on the future evolution of the system is assessed through value function approximations. Computational experiments indicate that the strategies we present yield high-quality solutions. We also present comparisons with conventional stochastic programming methods.

Keywords dynamic programming; approximate dynamic programming; stochastic approximation; large-scale optimization

1. Introduction

Many problems in operations research can be posed as managing a set of resources over multiple time periods under uncertainty. The resources may take on different forms in different applications: vehicles and containers for fleet management, doctors and nurses for personnel scheduling, cash and stocks for financial planning. Similarly, the uncertainty may have different characterizations in different applications: load arrivals and weather conditions for fleet management, patient arrivals for personnel scheduling, interest rates for financial planning. Despite the differences in terminology and application domain, a unifying aspect of these problems is that we have to make decisions under the premise that the decisions we make now will affect the future evolution of the system, and the future evolution of the system is also affected by random factors beyond our control.

A classical approach for solving such problems is to use the theory of Markov decision processes. The fundamental idea is to use a state variable that represents all information relevant to the future evolution of the system. Given the current value of the state variable, value functions capture the total expected cost incurred by the system over the whole planning horizon. Unfortunately, time and storage requirements for computing the value functions through conventional approaches, such as value iteration and policy iteration, increase exponentially with the number of dimensions of the state variable. For the applications above, these conventional approaches are simply intractable.

This chapter presents a modeling framework for large-scale resource allocation problems, along with a fairly flexible algorithmic framework that can be used to obtain good solutions for them. Our modeling framework is motivated by transportation applications, but it

provides enough generality to capture a variety of other problem settings. We do not focus on a specific application domain throughout the chapter, although we use the transportation setting to give concrete examples. The idea behind our algorithmic framework is to formulate the problem as a dynamic program and to use tractable approximations of the value functions, which are obtained by using simulated trajectories of the system and iteratively improving on (possibly naive) initial value function approximations.

The organization of the chapter is as follows. Sections 2 and 3 respectively present our modeling and algorithmic frameworks for describing and solving resource allocation problems. Section 4 describes a variety of methods that one can use to improve on the initial value function approximations. Section 5 focuses on the stepsize choices for the methods described in §4. In §6, we review other possible approaches for solving resource allocation problems, most of which are motivated by the field of stochastic programming. Section 7 presents some computational experiments. We conclude in §8 with possible extensions and unresolved issues.

2. Modeling Framework

This section describes a modeling framework for resource allocation problems. Our approach borrows ideas from mathematical programming, probability theory, and computer science. This modeling framework has been beneficial to us for several reasons. First, it offers a modeling language independent of the problem domain; one can use essentially the same language to describe a problem that involves assigning trucks to loads or a problem that involves scheduling computing tasks on multiple servers. Second, it extensively uses terminology—such as resources, decisions, transformation, and information—familiar to nonspecialists. This enables us to use our modeling framework as a communication tool when talking to a variety of people. Third, it is software-friendly; the components of our modeling framework can easily be mapped to software objects. This opens the door for developing general purpose software that can handle a variety of resource allocation problems.

We present our modeling framework by summarizing the major elements of a Markov decision process, ending with a formal statement of our objective function. However, working with this objective function is computationally intractable, and we focus on an approximation strategy in §3.

2.1. Modeling Time

Perhaps one of the most subtle dimensions of modeling a stochastic optimization problem is the modeling of time. In a stochastic model of a resource allocation problem, two processes are taking place: the flow of physical resources and the flow of information. The flow of information can be further divided into the flow of exogenous information and the flow of decisions.

For computational reasons, we assume that decisions are made at discrete points in time. These points in time, known as decision epochs, might be once every week, once every four hours, or once every second. They may also be determined by exogenous events, such as phone calls or arrivals of customers, in which case the time interval between the decision epochs is not constant.

In contrast, the arrival of exogenous information and the movement of resources occurs in continuous time. We might, for example, approximate a transportation problem by assuming that the decisions are made once every four hours, but the actual movements of the physical resources still occur in continuous time between the decision epochs. It is notationally convenient to represent the decision epochs with the integers $\mathcal{T} = \{0, 1, \ldots, T\}$ where T is the end of our planning horizon. Physical activities—such as arrivals of customers, departures of aircraft, job completions—and the arrival of information—such as customer requests, equipment failures, notifications of delays—can occur at continuous points in time between these decision epochs.

2.2. Resources

We use a fairly general notation to model resources, which handles both simple resources—such as oil, money, agricultural commodities—and complex resources—such as people, specialized machinery. We represent resources using

$\mathcal{A}$ = Attribute space of the resources. We usually use a to denote a generic element of the attribute space and refer to $a = (a_1, a_2, \ldots, a_I)$ as an attribute vector.
R_{ta} = Number of resources with attribute vector a at time period t just before a decision is made.
$R_t = (R_{ta})_{a \in \mathcal{A}}$.

Roughly speaking, the attribute space represents the set of all possible states of a particular resource. For example, letting $\mathcal{I}$ be the set of locations in the transportation network and $\mathcal{V}$ be the set of vehicle types, and assuming that the maximum travel time between any origin-destination pair is τ time periods, the attribute space of the vehicles in the fleet-management setting is $\mathcal{A} = \mathcal{I} \times \{0, 1, \ldots, \tau\} \times \mathcal{V}$. A vehicle with the attribute vector

$$a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \text{inbound/current location} \\ \text{time to reach inbound location} \\ \text{vehicle type} \end{bmatrix} \tag{1}$$

is a vehicle of type a_3 that is inbound to (or at) location a_1 and that will reach location a_1 at time a_2 (it is in the attribute a_2 that we model time continuously). The attribute a_2 might also be the time remaining until the vehicle is expected to arrive, or it might even be the departure time from the origin (this might be needed if the travel time is random). We note that certain attributes can be dynamic, such as inbound/current location, and certain attributes can be static, such as vehicle type. We access the number of vehicles with attribute vector a at time period t by referring to R_{ta}. This implies that we can "put" the vehicles with the same attribute vector in the same "bucket" and treat them as indistinguishable.

We assume that our resources are being used to serve demands; for example, demands for finishing a job, moving a passenger, or carrying a load of freight. We model the demands using

$\mathcal{B}$ = Attribute space of the demands. We usually use b to denote a generic element of the attribute space.
D_{tb} = Number of demands with attribute vector b waiting to be served at time period t.
$D_t = (D_{tb})_{b \in \mathcal{B}}$.

To keep the notation simple, we assume that any unserved demands are immediately lost.

Although we mostly consider the case where the resources are indivisible and R_{ta} takes integer values, R_{ta} may be allowed to take fractional values. For example, R_{ta} may represent the inventory level of a certain type of product at time period t measured in kilograms. Also, we mostly consider the case where the attribute space is finite. Finally, the definition of the attribute space implies that the resources we are managing are uniform; that is, the attribute vector for each resource takes values in the same space. However, by defining multiple attribute spaces, say $\mathcal{A}^1, \ldots, \mathcal{A}^N$, we can deal with multiple types of resources. For example, $\mathcal{A}^1$ may correspond to the drivers, whereas $\mathcal{A}^2$ may correspond to the trucks.

The attribute vector is a flexible object that allows us to model a variety of situations. In the fleet-management setting with single-period travel times and a homogenous fleet, the attribute space is as simple as $\mathcal{I}$. On the other extreme, we may be dealing with vehicles

with the attribute vector

$$\begin{bmatrix} \text{inbound/current location} \\ \text{time to reach inbound location} \\ \text{duty time within shift} \\ \text{days away from home} \\ \text{vehicle type} \\ \text{home domicile} \end{bmatrix}. \tag{2}$$

Based on the nature of the attribute space, we can model a variety of well-known problem classes.

1. *Single-product inventory control problems.* If the attribute space is a singleton, say $\{a\}$, then R_{ta} simply gives the inventory count at time period t.

2. *Multiple-product inventory control problems.* If we have $\mathcal{A} = \{1, \ldots, N\}$ and the attributes of the resources are static (product type), then R_{ta} gives the inventory count for product type a at time period t.

3. *Single-commodity min-cost network flow problems.* If we have $\mathcal{A} = \{1, \ldots, N\}$ and the attributes of the resources are dynamic, then R_{ta} gives the number of resources in state a at time period t. For example, this type of a situation arises when one manages a homogenous fleet of vehicles whose only attributes of interest are their locations. Our terminology is motivated by the fact that the deterministic versions of these problems can be formulated as min-cost network flow problems.

4. *Multicommodity min-cost network flow problems.* If we have $\mathcal{A} = \{1, \ldots, I\} \times \{1, \ldots, K\}$, and the first element of the attribute vector is static and the second element is dynamic, then $R_{t,[i,k]}$ gives the number of resources of type i that are in state k at time period t. For example, this situation type arises when one manages a heterogeneous fleet of vehicles whose only attributes of interest are their sizes (i) and locations (k).

5. *Heterogeneous resource allocation problems.* This is a generalization of the previous problem class in which the attribute space involves more than two dimensions, some static and some dynamic.

From a purely mathematical viewpoint, because we can "lump" all information about a resource into one dynamic attribute, single-commodity min-cost network flow problems provide enough generality to capture the other four problem classes. However, from the algorithmic viewpoint, the solution methodology we use and our ability to obtain integer solutions depend very much on what problem class we work. For example, we can easily enumerate all possible attribute vectors in $\mathcal{A}$ for the first four problem classes, but this may not be possible for the last problem class. When obtaining integer solutions is an issue, we often exploit a network flow structure. This may be possible for the first three problem classes, but not for the last two.

We emphasize that the attribute space is different than what is commonly referred to as the state space in Markov decision processes. The attribute space represents the set of all possible states of a particular resource. On the other hand, the state space in Markov decision processes refers to the set of all possible values that the resource state vector R_t can take. For example, in the fleet-management setting, the number of elements of the attribute space $\mathcal{A} = \mathcal{I} \times \{0, 1, \ldots \tau\} \times \mathcal{V}$ is on the order of several thousands. On the other hand, the state space includes all possible allocations of the fleet among different locations—an intractable number even for problems with small numbers of vehicles in the fleet, locations, and vehicle types.

2.3. Evolution of Information

We define

$$\begin{aligned} \widehat{R}_{ta}(R_t) &= \text{Random variable representing the change in the number of resources with} \\ &\qquad \text{attribute vector } a \text{ that occurs during time period } t. \\ \widehat{R}_t(R_t) &= (\widehat{R}_{ta}(R_t))_{a \in \mathcal{A}}. \end{aligned}$$

The random changes in the resource state vector may occur due to new resource arrivals or changes in the status of the existing resources. For notational brevity, we usually suppress the dependence on R_t. We model the flow of demands in a similar way by defining

$\widehat{D}_{tb}(R_t)$ = Random variable representing the new demands with attribute vector b that become available during time period t.
$\widehat{D}_t(R_t) = (\widehat{D}_{tb}(R_t))_{b\in\mathcal{B}}$.

From time to time, we need a generic variable to represent all the exogenous information that become available during time period t. The research community has not adopted a standard notation for exogenous information; we use

$$W_t = \text{Exogenous information that become available during time period } t.$$

For our problem, we have $W_t = (\widehat{R}_t, \widehat{D}_t)$.

2.4. The State Vector

The state vector captures the information we need at a certain time period to model the future evolution of the system. Virtually every textbook on dynamic programming represents the state vector as the information available just before we make the decisions. If we let S_t be the state of the system just before we make the decisions at time period t, then we have

$$S_t = (R_t, D_t).$$

We refer to S_t as the predecision state vector to emphasize that it is the state of the system just before we make the decisions at time period t. To simplify our presentation, we assume that any unserved demands are lost, which means that $D_t = \widehat{D}_t$. We will also find it useful to use the state of the system immediately after we make the decisions. We let

R_t^x = The resource state vector immediately after we make the decisions at time period t.

Because we assume that any unserved demands are lost, the state of the system immediately after we make the decisions at time period t is given by

$$S_t^x = R_t^x.$$

We refer to S_t^x as the postdecision state vector. For notational clarity, we often use R_t^x to capture the postdecision state vector.

It helps to summarize the sequence of states, decisions, and information by using

$$(S_0, x_0, S_0^x, W_1, S_1, x_1, S_1^x, \ldots, W_t, S_t, x_t, S_t^x, \ldots, W_T, S_T, x_T, S_T^x),$$

where x_t is the decision vector at time period t.

2.5. Decisions

Decisions are the means by which we can modify the attributes of the resources. We represent the decisions by defining

$\mathcal{C}$ = Set of decision classes. We can capture a broad range of resource allocation problems by using two classes of decisions; D to serve a demand and M to modify a resource without serving a demand.
$\mathcal{D}^D$ = Set of decisions to serve a demand. Each element of $\mathcal{D}^D$ represents a decision to serve a demand with a particular attribute vector; that is, there is an attribute vector $b_d \in \mathcal{B}$ for each $d \in \mathcal{D}^D$.

$\mathcal{D}^M$ = Set of decisions to modify a resource without serving a demand. In the transportation setting, this often refers to moving a vehicle from one location to another, but it can also refer to repairing the vehicle or changing its configuration. We assume that one element of $\mathcal{D}^M$ is a decision that represents "doing nothing."

$\mathcal{D} = \mathcal{D}^D \cup \mathcal{D}^M$.

x_{tad} = Number of resources with attribute vector a that are modified by using decision d at time period t.

c_{tad} = Profit contribution from modifying one resource with attribute vector a by using decision d at time period t.

Using standard terminology, $x_t = (x_{tad})_{a\in\mathcal{A},\, d\in\mathcal{D}}$ is the decision vector at time period t, along with the objective coefficients $c_t = (c_{tad})_{a\in\mathcal{A},\, d\in\mathcal{D}}$. If it is infeasible to apply decision d on a resource with attribute vector a, then we capture this by letting $c_{tad} = -\infty$. Fractional values may be allowed for x_{tad}, but we mostly consider the case where x_{tad} takes integer values.

In this case, the resource conservation constraints can be written as

$$\sum_{d\in\mathcal{D}} x_{tad} = R_{ta} \quad \text{for all } a \in \mathcal{A}. \tag{3}$$

These constraints simply state that the total number of resources with attribute vector a modified by using a decision at time period t equals the number of resources with attribute vector a.

Typically, there is a reward for serving a demand, but the number of such decisions is restricted by the number of demands. Noting that $d \in \mathcal{D}^D$ represents a decision to serve a demand with attribute vector b_d, we write the demand availability constraints as

$$\sum_{a\in\mathcal{A}} x_{tad} \le \widehat{D}_{t,\,b_d} \quad \text{for all } d \in \mathcal{D}^D.$$

We can now write our set of feasible decisions as

$$\mathcal{X}(S_t) = \Bigg\{ x_t \colon \sum_{d\in\mathcal{D}} x_{tad} = R_{ta} \text{ for all } a \in \mathcal{A} \tag{4}$$

$$\sum_{a\in\mathcal{A}} x_{tad} \le \widehat{D}_{t,\,b_d} \text{ for all } d \in \mathcal{D}^D \tag{5}$$

$$x_{tad} \in \mathbb{Z}_+ \text{ for all } a \in \mathcal{A},\ d \in \mathcal{D} \Bigg\}. \tag{6}$$

Our challenge is to find a policy or decision function that determines what decisions we should take. We let

$X_t^\pi(\cdot)$ = A function that maps the state vector S_t to the decision vector x_t at time period t; that is, we have $X_t^\pi(S_t) \in \mathcal{X}(S_t)$.

There can be many choices for this function; we focus on this issue in §3.

2.6. Transition Function

We capture the result of applying decision d on a resource with attribute vector a by

$$\delta_{a'}(a,d) = \begin{cases} 1 & \text{If applying decision } d \text{ on a resource with attribute vector } a \text{ transforms} \\ & \quad \text{the resource into a resource with attribute vector } a' \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Using the definition above, the resource dynamics can be written as

$$R^x_{ta} = \sum_{a' \in \mathcal{A}} \sum_{d \in \mathcal{D}} \delta_a(a', d)\, x_{ta'd} \quad \text{for all } a \in \mathcal{A}$$

$$R_{t+1,\,a} = R^x_{ta} + \widehat{R}_{t+1,\,a} \qquad \text{for all } a \in \mathcal{A}. \tag{8}$$

It is often useful to represent the system dynamics generically using

$$S_{t+1} = S^M(S_t, x_t, W_{t+1}),$$

where $W_{t+1} = (\widehat{R}_{t+1}, \widehat{D}_{t+1})$ is the new information arriving during time period $t+1$. Therefore, $S^M(\cdot,\cdot)$ is a function that maps the decision vector and the new information to a state vector for the next time period.

2.7. Objective Function

We are interested in finding decision functions $\{X^\pi_t(\cdot)\colon t \in \mathcal{T}\}$ that maximize the total expected profit contribution over the planning horizon. Noting that a set of decision functions $\{X^\pi_t(\cdot)\colon t \in \mathcal{T}\}$ define a policy π and letting Π be the set of all possible policies, we want to solve

$$\max_{\pi \in \Pi} \mathbb{E}\left\{ \sum_{t \in \mathcal{T}} C_t(X^\pi_t(S_t)) \right\}, \tag{9}$$

where we let $C_t(x_t) = \sum_{a \in \mathcal{A}} \sum_{d \in \mathcal{D}} c_{tad}\, x_{tad}$ for notational brevity. The problem above is virtually impossible to solve directly. The remainder of this chapter focuses on describing how approximate dynamic programming can be used to find high-quality solutions to this problem.

3. An Algorithmic Framework for Approximate Dynamic Programming

It is well-known that an optimal policy that solves problem (9) satisfies the Bellman equation

$$V_t(S_t) = \max_{x_t \in \mathcal{X}(S_t)} C_t(x_t) + \mathbb{E}\{V_{t+1}(S^M(S_t, x_t, W_{t+1})) \,|\, S_t\}. \tag{10}$$

It is also well-known that solving problem (10) suffers from the so-called curse of dimensionality. It is typically assumed that we have to solve (10) for every possible value of the state vector S_t. When S_t is a high-dimensional vector, the number of possible values for S_t quickly becomes intractably large. For our problems, S_t may have hundreds of thousands of dimensions.

Unfortunately, the picture is worse than it seems at first sight; there are actually three curses of dimensionality. The first is the size of the state space, which explodes when S_t is a high-dimensional vector. The second is the size of the outcome space that becomes problematic when we try to compute the expectation in (10). This expectation is often hidden in the standard textbook representations of the Bellman equation, which is written as

$$V_t(S_t) = \max_{x_t \in \mathcal{X}(S_t)} C_t(x_t) + \sum_{s' \in \mathcal{S}} p(s' \,|\, S_t, x_t)\, V_{t+1}(s'),$$

where $\mathcal{S}$ is the set of all possible values for the state vector S_{t+1}, and $p(s' \,|\, S_t, x_t)$ is the probability that $S^M(S_t, x_t, W_{t+1}) = s'$ conditional on S_t and x_t. Most textbooks on dynamic programming assume that the transition probability $p(s' \,|\, S_t, x_t)$ is given, but in many problems such as ours, it can be extremely difficult to compute.

The third curse of dimensionality is the size of the action space $\mathcal{X}(S_t)$, which we refer to as the feasible region. Classical treatments of dynamic programming assume that we enumerate all possible elements of $\mathcal{X}(S_t)$ when solving problem (10). When x_t is a high-dimensional vector, this is again intractable.

3.1. An Approximation Strategy Using the Postdecision State Vector

The standard version of the Bellman equation in (10) is formulated using the predecision state vector. If we write the Bellman equation around the postdecision state vector R^x_{t-1}, then we obtain

$$V^x_{t-1}(R^x_{t-1}) = \mathbb{E}\left\{ \max_{x_t \in \mathcal{X}(R^x_{t-1}, \widehat{R}_t, \widehat{D}_t)} C_t(x_t) + V^x_t(S^{M,x}(S_t, x_t)) \;\middle|\; R^x_{t-1} \right\}, \tag{11}$$

where we use the function $S^{M,x}(\cdot)$ to capture the dynamics of the postdecision state vector given in (8); that is, we have $R^x_t = S^{M,x}(S_t, x_t)$.

Not surprisingly, problem (11) is also computationally intractable. However, we can drop the expectation to write

$$\widetilde{V}^x_{t-1}(R^x_{t-1}, \widehat{R}_t, \widehat{D}_t) = \max_{x_t \in \mathcal{X}(R^x_{t-1}, \widehat{R}_t, \widehat{D}_t)} C_t(x_t) + V^x_t(S^{M,x}(R^x_{t-1}, W_t(\omega), x_t)), \tag{12}$$

where $W_t(\omega) = (\widehat{R}_t, \widehat{D}_t)$ is a sample realization of the new information that arrived during time interval t. The term $\widetilde{V}^x_{t-1}(S^x_{t-1}, \widehat{R}_t, \widehat{D}_t)$ is a place holder. Rather than computing the expectation, we solve the problem above for a particular realization of $(\widehat{R}_t, \widehat{D}_t)$; that is, given R^x_{t-1} and $(\widehat{R}_t, \widehat{D}_t)$, we compute a single decision x_t. Therefore, we can solve the second curse of dimensionality that arises due to the size of the outcome space by using the postdecision state vector.

However, we still do not know the value function $V^x_t(\cdot)$. To overcome this problem, we replace the value function with an approximation that we denote by using $\overline{V}^x_t(\cdot)$. In this case, our decision function is to solve the problem

$$X^\pi_t(R^x_{t-1}, \widehat{R}_t, \widehat{D}_t) = \operatorname*{arg\,max}_{x_t \in \mathcal{X}(R^x_{t-1}, \widehat{R}_t, \widehat{D}_t)} C_t(x_t) + \overline{V}^x_t(S^{M,x}(S_t, x_t)). \tag{13}$$

Therefore, we solve the first curse of dimensionality arising from the size of the state space by using approximations of the value function. Finally, we pay attention to use specially structured value function approximations so that the problem above can be solved by using standard optimization techniques. This solves the third curse of dimensionality arising from the size of the action space.

Table 1. An algorithmic framework for approximate dynamic programming.

Step 1. Choose initial value function approximations, say $\{\overline{V}^{0,x}_t(\cdot)\colon t \in \mathcal{T}\}$. Initialize the iteration counter by letting $n = 1$.

Step 2. Initialize the time period by letting $t = 0$. Initialize the state vector $R^{n,x}_0$ to reflect the initial state of the resources.

Step 3. Sample a realization of $(\widehat{R}_t, \widehat{D}_t)$, say $(\widehat{R}^n_t, \widehat{D}^n_t)$. Solve the problem

$$x^n_t = \operatorname*{arg\,max}_{x_t \in \mathcal{X}(R^{n,x}_{t-1}, \widehat{R}^n_t, \widehat{D}^n_t)} C_t(x_t) + \overline{V}^{n-1,x}_t(S^{M,x}(S_t, x_t)) \tag{14}$$

and let $R^{x,n}_t = S^{M,x}(S_t, x_t)$.

Step 4. Increase t by 1. If $t \le T$, then go to Step 3.

Step 5. Use the information obtained at iteration n to update the value function approximations. For the moment, we denote this by

$$\{\overline{V}^{n,x}_t(\cdot)\colon t \in \mathcal{T}\} = \text{Update}(\{\overline{V}^{n-1,x}_t(\cdot)\colon t \in \mathcal{T}\}, \{R^{n,x}_t\colon t \in \mathcal{T}\}, \{(\widehat{R}^n_t, \widehat{D}^n_t)\colon t \in \mathcal{T}\}),$$

where Update$(\cdot)$ can be viewed as a function that maps the value function approximations, the resource state vectors, and the new information at iteration n to the updated value function approximations.

Step 6. Increase n by 1 and go to Step 2.

3.2. Approximating the Value Function

Unless we are dealing with a problem with a very special structure, it is difficult to come up with good value function approximations. The approximate dynamic programming framework we propose solves problems of the form (13) for each time period t, and iteratively updates and improves the value function approximations. We describe this idea in Table 1. We note that solving problems of the form (14) for all $t \in \mathcal{T}$ is equivalent to simulating the behavior of the policy characterized by the value function approximations $\{\overline{V}_t^{n-1,x}(\cdot)\colon t \in \mathcal{T}\}$. In Table 1, we leave the structure of the value function approximations and the inner workings of the Update$(\cdot)$ function unspecified. Different strategies to fill in these two gaps potentially yield different approximate dynamic programming methods.

A generic structure for the value function approximations is

$$\overline{V}_t^x(R_t^x) = \sum_{f \in \mathcal{F}} \theta_{tf}\, \phi_f(R_t^x), \tag{15}$$

where $\{\phi_f(R_t^x)\colon f \in \mathcal{F}\}$ are often referred to as features because they capture the important characteristics of the resource state vector from the perspective of capturing the total expected profit contribution in the future. For example, if we are solving a resource allocation problem, a feature may be the number of resources with a particular attribute vector. By adjusting the parameters $\{\theta_{tf}\colon f \in \mathcal{F}\}$, we obtain different value function approximations. The choice of the functions $\{\phi_f(\cdot)\colon f \in \mathcal{F}\}$ requires some experimentation and some knowledge of the problem structure. However, for given $\{\phi_f(\cdot)\colon f \in \mathcal{F}\}$, there exist a variety of methods to set the values of the parameters $\{\theta_{tf}\colon f \in \mathcal{F}\}$ so that the value function approximation in (15) is a good approximation to the value function $V_t^x(\cdot)$.

For resource allocation problems, we further specialize the value function approximation structure in (15). In particular, we use separable value function approximations of the form

$$\overline{V}_t^x(R_t^x) = \sum_{a \in \mathcal{A}} \overline{V}_{ta}^x(R_{ta}^x), \tag{16}$$

where $\{\overline{V}_{ta}^x(\cdot)\colon a \in \mathcal{A}\}$ are one-dimensional functions. We focus on two cases.

1. *Linear value function approximations.* For these value function approximations, we have $\overline{V}_{ta}^x(R_{ta}^x) = \bar{v}_{ta}\, R_{ta}^x$, where $\bar{v}_{ta}$ are adjustable parameters. We use the notation $\{\bar{v}_{ta}\colon a \in \mathcal{A}\}$ for the adjustable parameters because this emphasizes we are representing the value function approximation $\overline{V}_t^x(\cdot)$, but $\{\bar{v}_{ta}\colon a \in \mathcal{A}\}$ are simply different representations of $\{\theta_{tf}\colon f \in \mathcal{F}\}$ in (15).

2. *Piecewise-linear value function approximations.* These value function approximations assume that $\overline{V}_{ta}^x(\cdot)$ is a piecewise-linear concave function with points of nondifferentiability being subset of positive integers. In this case, letting Q be an upper bound on the total number of resources one can have at any time period, we can characterize $\overline{V}_{ta}^x(\cdot)$ by a sequence of numbers $\{\bar{v}_{ta}(q)\colon q = 1, \ldots, Q\}$, where $\bar{v}_{ta}(q)$ is the slope of $\overline{V}_{ta}^x(\cdot)$ over the interval $(q-1, q)$; that is, we have $\bar{v}_{ta}(q) = \overline{V}_{ta}^x(q) - \overline{V}_{ta}^x(q-1)$. Because $\overline{V}_{ta}^x(\cdot)$ is concave, we have $\bar{v}_{ta}(1) \geq \bar{v}_{ta}(2) \geq \cdots \geq \bar{v}_{ta}(Q)$.

4. Monte Carlo Methods for Updating the Value Function Approximations

In this section, our goal is to propose alternatives for the Update$(\cdot)$ function in Step 5 in Table 1.

Whether we use linear or piecewise-linear value function approximations of the form $\overline{V}_t^{n,x}(R_t^x) = \sum_{a \in \mathcal{A}} \overline{V}_{ta}^{n,x}(R_{ta}^x)$, each of the functions $\{\overline{V}_{ta}^{n,x}(\cdot)\colon a \in \mathcal{A}\}$ is characterized either by a single slope (for the linear case) or by a sequence of slopes (for the piecewise-linear case). Using e_a to denote the $|\mathcal{A}|$-dimensional unit vector with a 1 in the element corresponding to

$a \in \mathcal{A}$, we would like to use $V_t^x(R_t^{n,x} + e_a) - V_t^x(R_t^{n,x})$ to update and improve the slopes that characterize the function $\overline{V}_{ta}^{n,x}(\cdot)$. However, this requires knowledge of the exact value function. Instead, letting $\widetilde{V}_t^{n,x}(R_t^{n,x}, \widehat{R}_t^n, \widehat{D}_t^n)$ be the optimal objective value of problem (14), we propose using

$$\vartheta_{ta}^n = \widetilde{V}_t^{n,x}(R_t^{n,x} + e_a, \widehat{R}_t^n, \widehat{D}_t^n) - \widetilde{V}_t^{n,x}(R_t^{n,x}, \widehat{R}_t^n, \widehat{D}_t^n). \tag{17}$$

We begin by describing a possible alternative for the Update(·) function when the value function approximations are linear. After that, we move on to piecewise-linear value function approximations.

4.1. Updating Linear Value Function Approximations

The method we use for updating the linear value function approximations is straightforward. Assuming that the value function approximation at iteration n is of the form $\overline{V}_t^{n,x}(R_t^x) = \sum_{a \in \mathcal{A}} \bar{v}_{ta}^n R_{ta}^x$, we let

$$\bar{v}_{ta}^n = [1 - \alpha_{n-1}]\, \bar{v}_{ta}^{n-1} + \alpha_{n-1}\, \vartheta_{ta}^n \tag{18}$$

for all $a \in \mathcal{A}$, where $\alpha_n \in [0,1]$ is the smoothing constant at iteration n. In this case, the value function approximation to be used at iteration $n+1$ is given by $\overline{V}_t^{n,x}(R_t^x) = \sum_{a \in \mathcal{A}} \bar{v}_{ta}^n R_{ta}^x$.

Linear value function approximations can be unstable, and experimental work shows that they do not perform as well as piecewise-linear value function approximations. Linear value function approximations are especially well suited to problems in which the resources managed are fairly complex, producing a very large attribute space. In these problems, we typically find that R_{ta}^x is 0 or 1 and using piecewise-linear value function approximations provides little value. In addition, linear value functions are much easier to work with and generally are a good starting point.

4.2. Updating Piecewise-Linear Value Function Approximations

We now assume that the value function approximation after iteration n is of the form $\overline{V}_t^{n,x}(R_t^x) = \sum_{a \in \mathcal{A}} \overline{V}_{ta}^{n,x}(R_{ta}^x)$, where each $\overline{V}_{ta}^{n,x}(\cdot)$ is a piecewise-linear concave function with points of nondifferentiability being a subset of positive integers. In particular, assuming that $\overline{V}_{ta}^{n,x}(0) = 0$ without loss of generality, we represent $\overline{V}_{ta}^{n,x}(\cdot)$ by a sequence of slopes $\{\bar{v}_{ta}^n(q)\colon q = 1, \ldots, Q\}$ as in §3.2, where we have $\bar{v}_{ta}^n(q) = \overline{V}_{ta}^{n,x}(q) - \overline{V}_{ta}^{n,x}(q-1)$. Concavity of $\overline{V}_{ta}^{n,x}(\cdot)$ implies that $\bar{v}_{ta}^n(1) \geq \bar{v}_{ta}^n(2) \geq \cdots \geq \bar{v}_{ta}^n(Q)$.

We update $\overline{V}_{ta}^{n,x}(\cdot)$ by letting

$$\theta_{ta}^n(q) = \begin{cases} [1 - \alpha_{n-1}]\, \bar{v}_{ta}^{n-1}(q) + \alpha_{n-1}\, \vartheta_{ta}^n & \text{if } q = R_{ta}^{n,x} + 1 \\ \bar{v}_{ta}^{n-1}(q) & \text{if } q \in \{1, \ldots, R_{ta}^{n,x}, R_{ta}^{n,x} + 2, \ldots, Q\}. \end{cases} \tag{19}$$

The expression above is similar to (18), but the smoothing operation applies only to the "relevant" part of the domain of $\overline{V}_{ta}^{n,x}(\cdot)$. However, we note that we may not have $\theta_{ta}^n(1) \geq \theta_{ta}^n(2) \geq \cdots \geq \theta_{ta}^n(Q)$, which implies that if we let $\overline{V}_{ta}^{n,x}(\cdot)$ be the piecewise-linear function characterized by the sequence of slopes $\theta_{ta}^n = \{\theta_{ta}^n(q)\colon q = 1, \ldots, Q\}$, then $\overline{V}_{ta}^{n,x}(\cdot)$ is not necessarily concave. To make sure that $\overline{V}_{ta}^{n,x}(\cdot)$ is concave, we choose a sequence of slopes $\bar{v}_{ta}^n = \{\bar{v}_{ta}^n(q)\colon q = 1, \ldots, Q\}$ such that $\bar{v}_{ta}^n$ and θ_{ta}^n are not too "far" from each other and the sequence of slopes $\bar{v}_{ta}^n$ satisfy $\bar{v}_{ta}^n(1) \geq \bar{v}_{ta}^n(2) \geq \cdots \geq \bar{v}_{ta}^n(Q)$. In this case, we let $\overline{V}_{ta}^{n,x}(\cdot)$ be the piecewise-linear concave function characterized by the sequence of slopes $\bar{v}_{ta}^n$.

There are several methods for choosing the sequence of slopes $\{\bar{v}_{ta}^n(q)\colon q = 1, \ldots, Q\}$. One possible method is to let $\bar{v}_{ta}^n$ be as follows

$$\begin{aligned} \bar{v}_{ta}^n = \arg\min \; & \sum_{q=1}^{Q} \left[z_q - \theta_{ta}^n(q)\right]^2 \\ \text{subject to} \quad & z_{q-1} - z_q \geq 0 \quad \text{for all } q = 2, \ldots, Q. \end{aligned} \tag{20}$$

Therefore, this method chooses the vector $\bar{v}_{ta}^n$ as the projection of the vector θ_{ta}^n onto the set $\mathcal{W} = \{z \in \mathbb{R}^Q : z_1 \geq z_2 \geq \cdots \geq z_Q\}$; that is, we have

$$\bar{v}_{ta}^n = \arg\min_{z \in \mathcal{W}} \|z - \theta_{ta}^n\|_2. \tag{21}$$

Using the Karush-Kuhn-Tucker conditions for problem (20), we can come up with a closed-form expression for the projection in (21). We only state the final result here. Because the vector θ_{ta}^n differs from the vector $\bar{v}_{ta}^n$ in one component and we have $\bar{v}_{ta}^n(1) \geq \bar{v}_{ta}^n(2) \geq \cdots \geq \bar{v}_{ta}^n(Q)$, there are three possible cases to consider; either $\theta_{ta}^n(1) \geq \theta_{ta}^n(2) \geq \cdots \geq \theta_{ta}^n(Q)$, or $\theta_{ta}^n(R_{ta}^{n,x}) < \theta_{ta}^n(R_{ta}^{n,x}+1)$, or $\theta_{ta}^n(R_{ta}^{n,x}+1) < \theta_{ta}^n(R_{ta}^{n,x}+2)$ should hold. If the first case holds, then we can choose $\bar{v}_{ta}^{n+1}$ in (21) as θ_{ta}^n, and we are done. If the second case holds, then we find the largest $q^* \in \{2, \ldots, R_{ta}^{n,x}+1\}$ such that

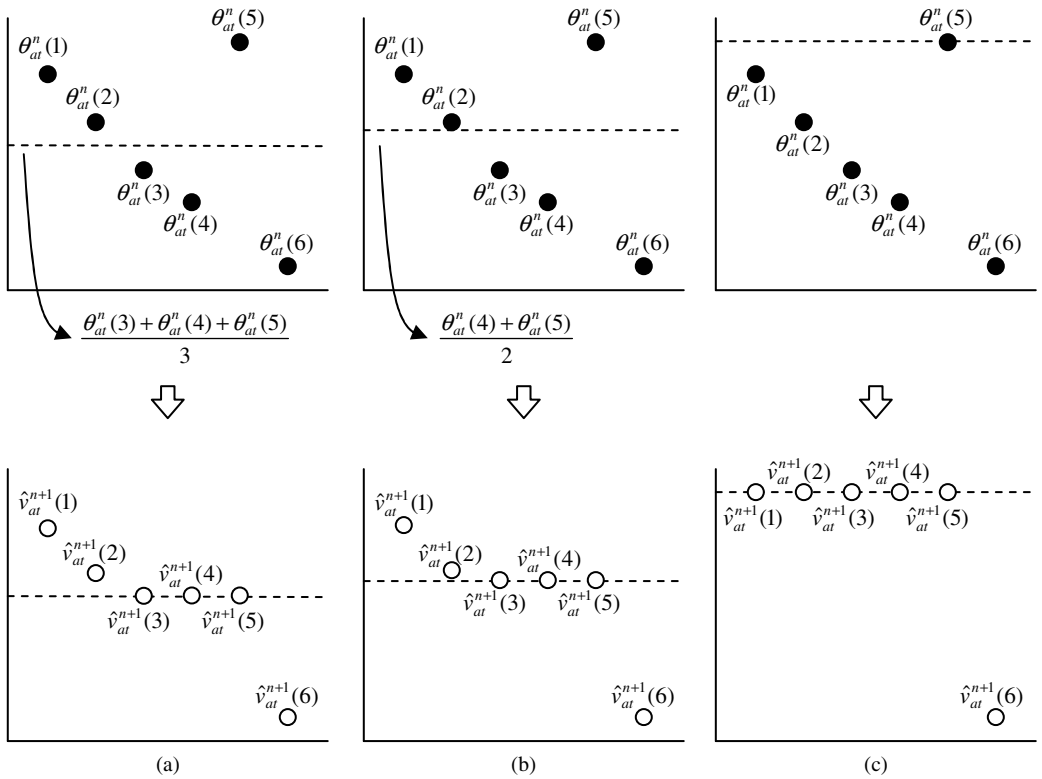
$$\theta_{ta}^n(q^*-1) \geq \frac{1}{R_{ta}^{n,x}+2-q^*} \sum_{q=q^*}^{R_{ta}^{n,x}+1} \theta_{ta}^n(q).$$

If such q^* cannot be found, then we let $q^* = 1$. It is straightforward to check that the vector $\bar{v}_{ta}^n$ given by

$$\bar{v}_{ta}^n(q) = \begin{cases} \dfrac{1}{R_{ta}^{n,x}+2-q^*} \displaystyle\sum_{q=q^*}^{R_{ta}^{n,x}+1} \theta_{ta}^n(q) & \text{if } q \in \{q^*, \ldots, R_{ta}^{n,x}+1\} \\ \theta_{ta}^n(q) & \text{if } q \notin \{q^*, \ldots, R_{ta}^{n,x}+1\} \end{cases} \tag{22}$$

which satisfies the Karush-Kuhn-Tucker conditions for problem (20). If the third case holds, then one can apply a similar argument. Figure 1a shows how this method works. The black

FIGURE 1. Three possible methods for choosing the vector $\bar{v}_{ta}^n$.



Note. In this figure, we assume that $Q = 6$, $R_{ta}^{n,x}+1 = 5$ and $q^* = 3$.

circles in the top portion of this figure show the sequence of slopes $\{\theta^n_{ta}(q)\colon q = 1, \ldots, Q\}$, whereas the white circles in the bottom portion show the sequence of slopes $\{\bar{v}^n_{ta}(q)\colon q = 1, \ldots, Q\}$ computed through (22).

Recalling the three possible cases considered above, a second possible method first computes

$$M^* = \begin{cases} \theta^n_{ta}(R^{n,x}_{ta}+1) & \text{if } \theta^n_{ta}(1) \geq \theta^n_{ta}(2) \geq \cdots \geq \theta^n_{ta}(Q) \\ \dfrac{\theta^n_{ta}(R^{n,x}_{ta}) + \theta^n_{ta}(R^{n,x}_{ta}+1)}{2} & \text{if } \theta^n_{ta}(R^{n,x}_{ta}) < \theta^n_{ta}(R^{n,x}_{ta}+1) \\ \dfrac{\theta^n_{ta}(R^{n,x}_{ta}+1) + \theta^n_{ta}(R^{n,x}_{ta}+2)}{2} & \text{if } \theta^n_{ta}(R^{n,x}_{ta}+1) < \theta^n_{ta}(R^{n,x}_{ta}+2), \end{cases} \tag{23}$$

and lets

$$\bar{v}^n_{ta}(q) = \begin{cases} \max\{\theta^n_{ta}(q),\, M^*\} & \text{if } q \in \{1, \ldots, R^{n,x}_{ta}\} \\ M^* & \text{if } q = R^{n,x}_{ta}+1 \\ \min\{\theta^n_{ta}(q),\, M^*\} & \text{if } q \in \{R^{n,x}_{ta}+2, \ldots, Q\}. \end{cases} \tag{24}$$

Interestingly, it can be shown that (23) and (24) are equivalent to letting

$$\bar{v}^{n+1}_{ta} = \arg\min_{z \in \mathcal{W}} \ \|z - \theta^n_{ta}\|_\infty.$$

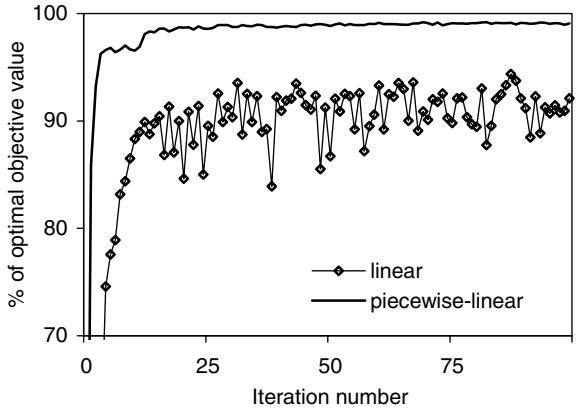
Therefore, the first method is based on a Euclidean-norm projection, whereas the second method is based on a max-norm projection. Figure 1b shows how this method works.

A slight variation on the second method yields a third method, which computes $M^* = \theta^n_{ta}(R^{n,x}_{ta}+1)$ and lets the vector $\bar{v}^n_{ta}$ be as in (24). This method does not have an interpretation as a projection. Figure 1c shows how this method works.

There are convergence results for the three methods described above. All of these results are in limited settings that assume that the planning horizon contains two time periods and the state vector is one-dimensional. Roughly speaking, they show that if the state vector $R^{n,x}_1$ generated by the algorithmic framework in Table 1 satisfies $\sum_{n=1}^{\infty} \mathbf{1}(R^{n,x}_1 = q) = \infty$ with probability 1 for all $q = 1, \ldots, Q$ and we use one of the three methods described above to update the piecewise-linear value function approximations, then we have $\lim_{n\to\infty} \bar{v}^n_1(R^x_1) = V_1(R^x_1) - V_1(R^x_1 - 1)$ for all $R^x_1 = 1, \ldots, Q$ with probability 1. Throughout, we omit the subscript a because the state vector is one-dimensional and use $\mathbf{1}(\cdot)$ to denote the indicator function. When we apply these methods to large resource allocation problems with multidimensional state vectors, they are only approximate methods that seem to perform quite well in practice.

Experimental work indicates that piecewise-linear value function approximations can provide better objective values and more stable behavior than linear value function approximations. Figure 2 shows the performances of linear and piecewise-linear value function approximations on a resource allocation problem with deterministic data. The horizontal axis is the iteration number in the algorithmic framework in Table 1. The vertical axis is the performance of the policy obtained at a particular iteration, expressed as a percentage of the optimal objective value. We obtain the optimal objective value by formulating the problem as a large integer program. Figure 2 shows that the policies characterized by piecewise-linear value function approximations may perform almost as well as the optimal solution, whereas the policies characterized by linear value function approximations lag behind significantly. Furthermore, the performances of the policies characterized by linear value function approximations at different iterations can fluctuate. Nevertheless, linear value function approximations may be used as prototypes before moving on to more-sophisticated approximation strategies, or we may have to live with them simply because the resource allocation problem we are dealing with is too complex.

FIGURE 2. Performances of linear and piecewise-linear value function approximations on a resource allocation problem with deterministic data.



5. Stepsizes

Approximate dynamic programming depends heavily on using information from the latest iteration to update a value function approximation. This results in updates of the form

$$\bar{v}_{ta}^{n} = [1 - \alpha_{n-1}]\ \bar{v}_{ta}^{n-1} + \alpha_{n-1}\, \vartheta_{ta}^{n}, \tag{25}$$

where α_{n-1} is the stepsize used in iteration n. This intuitive updating formula is known variously as exponential smoothing, a linear filter, or a stochastic approximation procedure. The equation actually comes from the optimization problem

$$\min_{\theta}\ \mathbb{E}\{F(\theta, \widehat{R})\},$$

where $F(\theta, \widehat{R})$ is a function of θ and random variable $\widehat{R}$. Furthermore, we assume that we cannot compute the expectation either because the function is too complicated or because we do not know the distribution of $\widehat{R}$. We can still solve the problem using an algorithm of the form

$$\theta^{n} = \theta^{n-1} - \alpha_{n-1}\, \nabla F(\theta^{n-1}, \widehat{R}^{n}), \tag{26}$$

where θ^{n-1} is our estimate of the optimal solution after iteration $n-1$, and $\widehat{R}^{n}$ is a sample of the random variable $\widehat{R}$ at iteration n. If $F(\cdot, \widehat{R}^{n})$ is not differentiable, then we assume that $\nabla F(\theta^{n-1}, \widehat{R}^{n})$ is a subgradient of the function. The updating in (26) is known as a stochastic gradient algorithm, because we are taking a gradient of $F(\cdot, \widehat{R}^{n})$ with respect to θ at a sample realization of the random variable $\widehat{R}$.

Assume that our problem is to estimate the mean of the random variable $\widehat{R}$. We assume that the distribution of the random variable $\widehat{R}$ is unknown, but we can obtain samples $\widehat{R}^{1}, \widehat{R}^{2}, \ldots$. Since we have $\mathbb{E}\{\widehat{R}\} = \arg\min_{\theta} \mathbb{E}\{(\theta - \widehat{R})^2\}$, a reasonable approach is to let

$$F(\theta, \widehat{R}) = \frac{1}{2}(\theta - \widehat{R})^2$$

and use (26). Letting θ^{n} be the estimate of $\mathbb{E}\{\widehat{R}\}$ obtained after iteration n, since we have $\nabla F(\theta, \widehat{R}) = (\theta - \widehat{R})$, we obtain

$$\begin{aligned}\theta^{n} &= \theta^{n-1} - \alpha_{n-1}\, \nabla F(\theta^{n-1}, \widehat{R}^{n}) \\ &= \theta^{n-1} - \alpha_{n-1}\, (\theta^{n-1} - \widehat{R}^{n}) = (1 - \alpha_{n-1})\, \theta^{n-1} + \alpha_{n-1}\, \widehat{R}^{n}.\end{aligned}$$

Among the last two equalities above, the first has the same form as the stochastic gradient algorithm and the second has the same form as exponential smoothing.

There is an elegant theory that tells us this method works, but there are some simple restrictions on the stepsizes. In addition to the requirement that $\alpha_{n-1} \geq 0$ for $n = 1, 2, \ldots$, the stepsizes must also satisfy

$$\sum_{n=1}^{\infty} \alpha_{n-1} = \infty \quad \sum_{n=1}^{\infty} (\alpha_{n-1})^2 < \infty.$$

The first condition ensures that the stepsizes do not decline too quickly; otherwise, the algorithm may stall out prematurely. The second ensures that they do not decline too slowly, which ensures that the algorithm actually converges in the limit. One stepsize rule that satisfies this condition is $\alpha_{n-1} = 1/(n-1)$. This rule is special because it produces a simple averaging of all the observations, which is to say that

$$\theta^n = \frac{1}{n} \sum_{m=1}^{n} \widehat{R}^m.$$

If we are getting a series of observations of $\widehat{R}$ from a stationary distribution, this would be fine; in fact, this is the best we can do. However, in dynamic programming, our updates of the value function are changing over the iterations as we try to converge on an optimal policy. As a result, the values ϑ_{ta}^n are coming from a distribution that is changing over the iterations. For this reason, it is well known that the so-called "$1/n$" stepsize rule produces stepsizes that decline much too quickly.

A variety of strategies have evolved over the years to counter this effect. One fairly general class of formulas is captured by

$$\alpha_n = \begin{cases} \alpha_0 & \text{if } n = 0 \\ \alpha_0 \dfrac{b/n + a}{b/n + a + n^{\beta} - 1} & \text{if } n > 0. \end{cases}$$

If $b = 0$, $\alpha_0 = 1$, $\beta = 1$, and $a = 1$, then we obtain the "$1/n$" stepsize rule. As a is increased (values in the 5 to 20 range work quite well) or β is decreased (for theoretical reasons, it should stay above 0.5), the rate at which the stepsize decreases slows quite a bit. Raising the parameter b has the effect of keeping the stepsize very close to the initial value for a while before allowing the stepsize to decrease. This is useful for certain classes of delayed learning, where a number of iterations must occur before the system starts to obtain meaningful results. We have found that $a = 8$, $b = 0$, and $\beta = 0.7$ works quite well for many dynamic programming applications.

Another useful rule is McClain's formula, given by

$$\alpha_n = \begin{cases} \alpha_0 & \text{if } n = 0 \\ \dfrac{\alpha_{n-1}}{1 + \alpha_{n-1} - \bar{\alpha}} & \text{if } n \geq 1. \end{cases}$$

If $\bar{\alpha} = 0$ and $\alpha_0 = 1$, then this formula gives $\alpha_n = 1/n$. For $0 < \bar{\alpha} < 1$, the formula produces a sequence of decreasing stepsizes that initially behaves like $1/n$, but decreases to $\bar{\alpha}$ instead of 0. This is a way of ensuring that the stepsize does not get too small.

The challenge with stepsizes is that if we are not careful, then we may design an algorithm that works poorly when, in fact, the only problem is the stepsize. It may be quite frustrating tuning the parameters of a stepsize formula; we may be estimating many thousands of parameters, and the best stepsize formula may be different for each parameter.

For this reason, researchers have studied a number of stochastic stepsize formulas. These are stepsize rules where the size of the stepsize depends on what is happening over the course of the algorithm. Because the stepsize at iteration n depends on the data, the stepsize itself

is a random variable. One of the earliest and most famous of the stochastic stepsize rules is known as Kesten's rule given by

$$\alpha_n = \alpha_0 \frac{a}{a + K^n}, \tag{27}$$

where α_0 is the initial stepsize and a is a parameter to be calibrated. Letting

$$\varepsilon^n = \theta^{n-1} - \widehat{R}^n$$

be the error between our previous estimate of the random variable and the latest observation, if θ^{n-1} is far from the true value, then we expect to see a series of errors with the same sign. The variable K^n counts the number of times that the sign of the error has changed by

$$K^n = \begin{cases} n & \text{if } n = 0, 1 \\ K^{n-1} + \mathbf{1}(\varepsilon^n \varepsilon^{n-1} < 0) & \text{otherwise.} \end{cases} \tag{28}$$

Thus, every time the sign changes, indicating that we are close to the optimal solution, the stepsize decreases.

Ideally, a stepsize formula should decline as the level of variability in the observations increase and should increase when the underlying signal is changing quickly. A formula that does this is

$$\alpha^n = 1 - \frac{\sigma^2}{(1 + \lambda^{n-1})\,\sigma^2 + (\beta^n)^2},$$

where

$$\lambda^n = \begin{cases} (\alpha_n)^2 & \text{if } n = 1 \\ (\alpha_n)^2 + (1 - \alpha_n)^2 \lambda^{n-1} & \text{if } n > 1. \end{cases}$$

In the expression above, σ^2 is the noise in the observations, and β^n is the difference between the true value and the estimated value, which we refer to as the bias. It can be shown that

TABLE 2. The optimal stepsize algorithm.

Step 0. Choose an initial estimate $\bar{\theta}^0$ and an initial stepsize α_0. Assign initial values to the parameters by letting $\bar{\beta}^0 = 0$ and $\bar{\delta}^0 = 0$. Choose an initial value for the error stepsize γ_0 and a target value for the error stepsize $\bar{\gamma}$. Set the iteration counter $n = 1$.

Step 1. Obtain the new observation $\widehat{R}^n$.

Step 2. Update the following parameters by letting

$$\gamma_n = \frac{\gamma_{n-1}}{1 + \gamma_{n-1} - \bar{\gamma}}$$

$$\bar{\beta}^n = (1 - \gamma_n)\,\bar{\beta}^{n-1} + \gamma_n(\widehat{R}^n - \bar{\theta}^{n-1})$$

$$\bar{\delta}^n = (1 - \gamma_n)\,\bar{\delta}^{n-1} + \gamma_n(\widehat{R}^n - \bar{\theta}^{n-1})^2$$

$$(\bar{\sigma}^n)^2 = \frac{\bar{\delta}^n - (\bar{\beta}^n)^2}{1 + \bar{\lambda}^{n-1}}.$$

Step 3. If $n > 1$, then evaluate the stepsizes for the current iteration by

$$\alpha_n = 1 - \frac{(\bar{\sigma}^n)^2}{\bar{\delta}^n}.$$

Step 4. Update the coefficient for the variance of the smoothed estimate by

$$\bar{\lambda}^n = \begin{cases} (\alpha_n)^2 & \text{if } n = 1 \\ (1 - \alpha_n)^2 \bar{\lambda}^{n-1} + (\alpha_n)^2 & \text{if } n > 1. \end{cases}$$

Step 5. Smooth the estimate by

$$\bar{\theta}^n = (1 - \alpha_{n-1})\,\bar{\theta}^{n-1} + \alpha_{n-1}\,\widehat{R}^n.$$

Step 6. If $\bar{\theta}^n$ satisfies some termination criterion, then stop. Otherwise, set $n = n + 1$ and go to Step 1.

if $\sigma^2 = 0$, then $\alpha_n = 1$, whereas if $\beta^n = 0$, then $\alpha_n = 1/n$. The problem is that neither of these quantities would normally be known; in particular, if we knew the bias, then it means we know the true value function.

Table 2 presents an adaptation of this formula for the case where the noise and bias are not known. This formula has been found to provide consistently good results for a broad range of problems, including those with delayed learning.

6. Other Approaches for Dynamic Resource Allocation Problems

To understand the relative simplicity of approximate dynamic programming and to provide benchmarks to measure solution quality, it is useful to review other methods for solving resource allocation problems.

6.1. A Deterministic Model

A common strategy employed to deal with randomness is to assume that the future random quantities take on their expected values and to formulate a deterministic optimization problem. For the resource allocation setting, this problem takes the form

$$\begin{aligned} \max \quad & \sum_{a\in\mathcal{A}} \sum_{d\in\mathcal{D}} c_{0ad}\, x_{0ad} && (29)\\ \text{subject to} \quad & \sum_{d\in\mathcal{D}} x_{0ad} = R_{0a} \quad \text{for all } a \in \mathcal{A} \\ & -\sum_{a'\in\mathcal{A}} \sum_{d\in\mathcal{D}} \delta_a(a', d)\, x_{0a'd} + \sum_{d\in\mathcal{D}} x_{0ad} = \mathbb{E}\{\widehat{R}_{ta}\} \quad \text{for all } a \in \mathcal{A} \\ & x_{0ad} \in \mathbb{Z}_+ \quad \text{for all } a \in \mathcal{A},\ d \in \mathcal{D}. \end{aligned}$$

It is important to keep in mind that the time at which flows happen is imbedded in the attribute vector. This makes for a very compact model, but one less transparent. In practice, we use problem (29) on a rolling horizon basis; we solve this problem to make the decisions at the first time period and implement these decisions. When it is time to make the decisions at the second time period, we solve a similar problem that involves the known resource state vector and the demands at the second time period.

Problem (29) uses only the expected values of the random quantities, disregarding the distribution information. However, there are certain applications, such as airline fleet assignment, where the uncertainty does not play a crucial role, and problem (29) can efficiently be solved as an integer multicommodity min-cost network flow problem.

6.2. Scenario-Based Stochastic Programming Methods

Stochastic programming emerges as a possible approach when one attempts to use the distribution information. In the remainder of this section, we review stochastic programming methods applicable to resource allocation problems. Thus far, we mostly focused on problems in which the decision variables take integer values. There has been much progress in the area of integer stochastic programming within the last decade, but, to our knowledge, there does not exist integer stochastic programming methods that can solve the resource allocation problems in the full generality that we present here. For this reason, we relax the integrality constraints throughout this section. To make the ideas transparent, we assume that the planning horizon contains two time periods, although most of the methods apply to problems with longer planning horizons.

Scenario-based stochastic programming methods assume that there exist a finite set of possible realizations for the random vector $(\widehat{R}_1, \widehat{D}_1)$, which we denote by $\{(\widehat{R}_1(\omega), \widehat{D}_1(\omega)): \omega \in \Omega\}$. In this case, using $p(\omega)$ to denote the probability of realization $(\widehat{R}_1(\omega), \widehat{D}_1(\omega))$, the

exact value function at the second time period can be computed by solving

$$V_0(R_0^x) = \max \sum_{\omega\in\Omega}\sum_{a\in\mathcal{A}}\sum_{d\in\mathcal{D}} p(\omega)\, c_{1ad}\, x_{1ad}(\omega) \tag{30}$$

$$\text{subject to} \quad \sum_{d\in\mathcal{D}} x_{1ad}(\omega) = R_{0a}^x + \widehat{R}_{1a}(\omega) \quad \text{for all } a\in\mathcal{A},\ \omega\in\Omega \tag{31}$$

$$\sum_{a\in\mathcal{A}} x_{1ad} \le \widehat{D}_{1,b_d}(\omega) \quad \text{for all } d\in\mathcal{D}^D,\ \omega\in\Omega,$$

where we omit the nonnegativity constraints for brevity. This approach allows complete generality in the correlation structure among the elements of the random vector $(\widehat{R}_1, \widehat{D}_1)$, but it assumes that this random vector is independent of R_1. Because the decision variables are $\{x_{1ad}(\omega)\colon a\in\mathcal{A},\ d\in\mathcal{D},\ \omega\in\Omega\}$, problem (30) can be large for practical applications.

6.3. Benders Decomposition-Based Methods

Because the resource state vector R_0^x appears on the right side of constraints (31), $V_0(R_0^x)$ is a piecewise-linear concave function of R_0^x. Benders decomposition-based methods refer to a class of methods that approximate the exact value function $V_0(\cdot)$ by a series of cuts that are constructed iteratively. In particular, letting $\{\lambda_1^i\colon i=1,\ldots,n-1\}$ and $\{\beta_{1a}^i\colon a\in\mathcal{A},\ i=1,\ldots,n-1\}$ be the sets of coefficients characterizing the cuts that have been constructed up to iteration n, the function

$$\overline{V}_0^n(R_0^x) = \min_{i\in\{1,\ldots,n-1\}} \lambda_1^i + \sum_{a\in\mathcal{A}} \beta_{1a}^i\, R_{0a}^x \tag{32}$$

is the approximation to the exact value function $V_0(\cdot)$ at iteration n. The details of how to generate the cuts are beyond our presentation.

6.4. Auxiliary Functions

As a last possible stochastic programming method, we describe an algorithm called the stochastic hybrid approximation procedure (SHAPE). This method is similar to the methods described in §4; it iteratively updates an approximation to the value function by using a formula similar to (18).

SHAPE uses value function approximations of the form

$$\overline{V}_0^{n,x}(R_0^x) = \overline{W}_0(R_0^x) + \sum_{a\in\mathcal{A}} \bar{v}_{0a}^n\, R_{0a}^x, \tag{33}$$

where $\overline{W}_0(\cdot)$ is a function specified in advance. In general, $\overline{W}_0(\cdot)$ is chosen so that it is easy to work with; for example, a polynomial. However, the procedure works best when $\overline{W}_0(\cdot)$ approximately captures the general shape of the value function. The second term on the right side of (33) is a linear value function approximation component that is adjusted iteratively. Consequently, the first nonlinear component of the value function approximation does not change over the iterations, but the second linear component is adjustable. We assume that $\overline{W}_0(\cdot)$ is a differentiable concave function with the gradient $\nabla\overline{W}_0(R_0^x) = (\nabla_a\overline{W}_0(R_0^x))_{a\in\mathcal{A}}$.

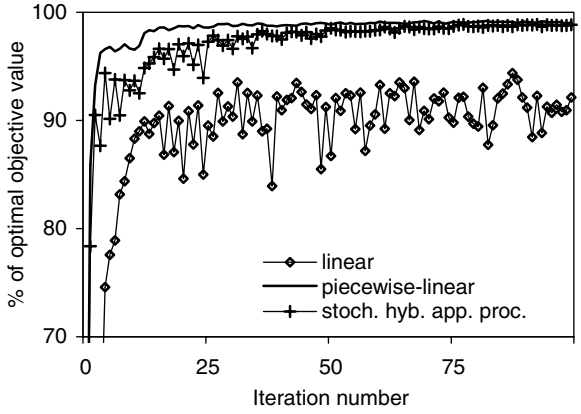
Using the value function approximation in (33), we first solve the approximate subproblem at the first time period to obtain

$$x_0^n = \arg\max_{x_0\in\mathcal{X}(R_0)} C_0(x_0) + \overline{V}_0^{n-1,x}(R^{M,x}(R_0,x_0)). \tag{34}$$

Letting $R_0^{n,x} = S^{M,x}(S_0,x_0^n)$ and $(\widehat{R}_1^n, \widehat{D}_1^n)$ be a sample of $(\widehat{R}_1, \widehat{D}_1)$, we then solve

$$\arg\max_{x_1\in\mathcal{X}(R_0^{n,x},\widehat{R}_1^n,\widehat{D}_1^n)} C_1(x_1).$$

FIGURE 3. Performances of SHAPE, and linear and piecewise-linear value function approximations.



In this case, using $\{\pi^n_{1a}: a \in \mathcal{A}\}$ to denote the optimal values of the dual variables associated with constraints (4) in the problem above, we let

$$\bar{v}^n_{0a} = [1 - \alpha_{n-1}]\, \bar{v}^{n-1}_{0a} + \alpha_{n-1}\, [\pi^n_{1a} - \nabla_a \overline{V}^{n-1}_0(R_0, x_0)],$$

where $\alpha_{n-1} \in [0,1]$ is the smoothing constant at iteration n. Therefore, the value function approximation at iteration n is given by $\overline{V}^{n,x}_0(R^x_0) = \overline{W}_0(R^x_0) + \sum_{a \in \mathcal{A}} \bar{v}^n_{0a}\, R^x_{0a}$. It is possible to show that this algorithm produces the optimal solution for two-period problems.

This method is simple to implement. Because we only update the linear component of the value function approximation, the structural properties of the value function approximation do not change. For example, if we choose $\overline{W}_0(\cdot)$ as a separable quadratic function, then the value function approximation is a separable quadratic function at every iteration. Nevertheless, SHAPE has not seen much attention from the perspective of practical implementations. The first reason for this is that $\overline{V}^{n,x}_0(\cdot)$ is a differentiable function, and the approximate subproblem in (34) is a smooth optimization problem. Given the surge in quadratic programming packages, we do not think this is a major issue anymore. The second reason is that the practical performance of the procedure can depend on the choice of $\overline{W}_0(\cdot)$, and there is no clear guideline for this choice. We believe that the methods described in §4 can be used for this purpose. We can use these methods to construct a piecewise-linear value function approximation, fit a strongly separable quadratic function to the piecewise-linear value function approximation, and use this fitted function for $\overline{W}_0(\cdot)$.

Figure 3 shows the performances of SHAPE, linear value function approximations, and piecewise-linear value function approximations on a resource allocation problem with deterministic data. The objective values obtained by SHAPE at the early iterations fluctuate, but they quickly stabilize, whereas the objective values obtained by linear value function approximations continue to fluctuate. The concave "auxiliary" function that SHAPE uses prevents the "bang-bang" behavior of linear value function approximations and provides more stable performance.

7. Computational Results

This section presents computational experiments on a variety of resource allocation problems. We begin by considering two-period problems and later move on to multiple-period problems. The primary reason we consider two-period problems is that there exists a variety of solution methods for them, some of which are described in §6, that we can use as benchmarks. This gives us a chance to carefully test the performance of the algorithmic framework in Table 1.

7.1. Two-Period Problems

In this section, we present computational experiments on two-period problems arising from the fleet-management setting. We assume that there is a single vehicle type and it takes one time period to move between any origin-destination pair. In this case, the attribute vector in (1) is of the form $a = [\text{inbound/current location}]$, and the attribute space $\mathcal{A}$ is simply the set of locations in the transportation network. There are two decision types with $\mathcal{C} = \{D, M\}$, where $\mathcal{D}^D$ and $\mathcal{D}^M$ have the same interpretations as in §2.5. We use piecewise-linear value function approximations and update them by using (19) and (20) with $\alpha_n = 20/(40 + n)$.

We generate a certain number of locations over a 100×100 region. At the beginning of the planning horizon, we spread the fleet uniformly over these locations. The loads between different origin-destination pairs and at different time periods are sampled from the Poisson distributions with the appropriate means. We focus on problems where the number of inbound loads to a particular location is negatively correlated with the number of outbound loads from that location. We expect that these problems require plenty of empty repositioning movements in their optimal solutions, and naive methods should not provide good solutions for them.

Evaluating the performances of the methods presented in this chapter requires two sets of iterations. In the first set, which we refer to as the training iterations, we follow the algorithmic framework in Table 1; we sample a realization of the random vector $(\widehat{R}_t, \widehat{D}_t)$ and solve problem (14) for each time period t, and update the value function approximations. In the second set, which we refer to as the testing iterations, we fix the value function approximations and simply simulate the behavior of the policy characterized by the value function approximations obtained during the training iterations. Consequently, the goal of the testing iterations is to test the quality of the value function approximations. For Benders decomposition-based methods, the training iterations construct the cuts that approximate the value functions, whereas the testing iterations simulate the behavior of the policy characterized by the cuts constructed during the training iterations. We vary the number of training iterations to see how fast we can obtain good policies through different methods. The particular version of Benders decomposition-based method that we use in our computational experiments is called cutting plane and partial sampling method. We henceforth refer to the approximate dynamic programming framework in Table 1 as ADP and cutting plane and partial sampling method as CUPPS.

For a test problem that involves 30 locations, Figure 4 shows the average objective values obtained in the testing iterations as a function of the number of training iterations. The white and gray bars in this figure, respectively, correspond to ADP and CUPPS. When the number of training iterations is relatively small, it appears that ADP provides better objective

FIGURE 4. Performances of ADP and CUPPS for different numbers of training iterations.

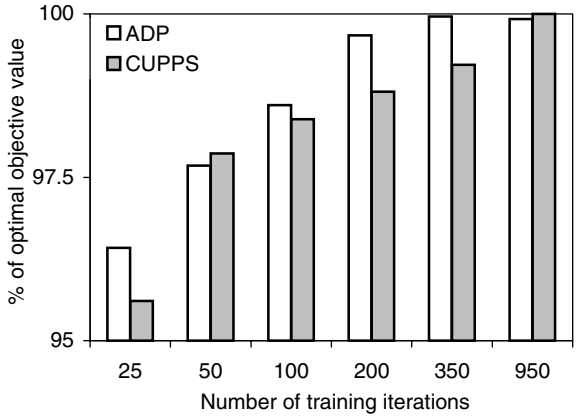
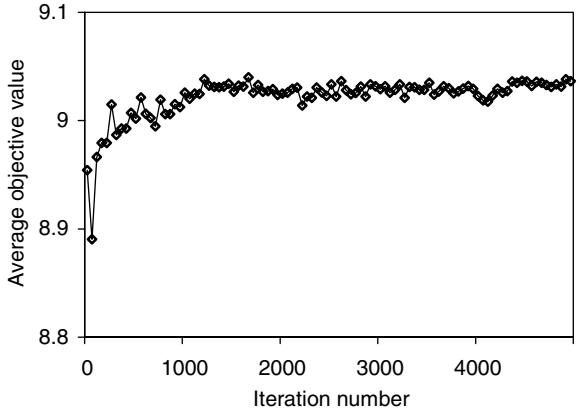


FIGURE 5. Performances of the policies obtained by ADP as a function of the number of training iterations.



values than CUPPS. Because CUPPS eventually solves the problem exactly and ADP is only an approximation strategy, if the number of training iterations is large, then CUPPS provides better objective values than ADP. Even after CUPPS obtains the optimal solution, the performance gap between ADP and CUPPS is a fraction of a percent. Furthermore, letting $\{\overline{V}_t^{n,x}(\cdot)\colon t \in \mathcal{T}\}$ be the set of value function approximations obtained by ADP at iteration n, Figure 5 shows the performance of the policy characterized by the value function approximations $\{\overline{V}_t^{n,x}(\cdot)\colon t \in \mathcal{T}\}$ as a function of the iteration number n. Performances of the policies stabilize after about 1,500 training iterations.

For test problems that involve different numbers of locations, Figure 6 shows the average objective values obtained in the testing iterations. In this figure, the number of training iterations is fixed at 200. For problems with few locations, the objective values obtained by ADP and CUPPS are very similar. As the number of locations grows, the objective values obtained by ADP are noticeably better than those obtained by CUPPS. The number of locations gives the number of dimensions of the value function. Therefore, for problems that involve high-dimensional value functions, it appears that ADP obtains good policies faster than CUPPS.

7.2. Multiperiod Problems

This section presents computational experiments on multiperiod problems arising from the fleet-management setting. To introduce some variety, we now assume that there are multiple vehicle and load types. In this case, the attribute space of the resources consists of vectors

FIGURE 6. Performances of ADP and CUPPS for problems with different numbers of locations.

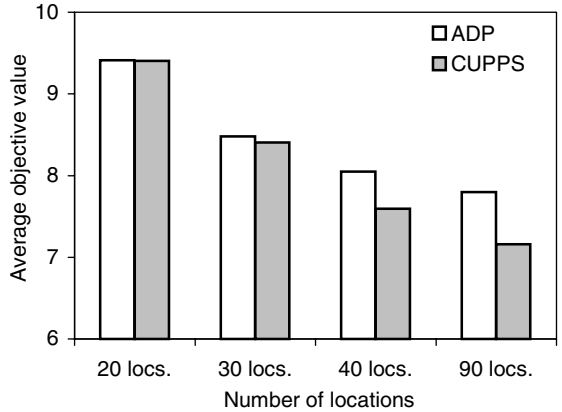


TABLE 3. Performance of ADP on different test problems.

Problem	(20,60,200)	(20,30,200)	(20,90,200)	(10,60,200)	(40,60,200)	(20,60,100)	(20,60,400)
% of opt. obj.val.	99.5	99.7	99.3	99.8	99.0	97.2	99.5

Note. The triplets denote the characteristics of the test problems, where the three elements are the number of locations, the number of time periods, and the fleet size.

of the form (1). We assume that we obtain a profit of $r\,D(o,d)\,C(l,v)$ when we use a vehicle of type v to carry a load of type l from location o to d, where r is the profit per mile, $D(o,d)$ is the distance between origin-destination pair (o,d), and $C(l,v) \in [0,1]$ captures the compatibility between load type l and vehicle type v. As $C(l,v)$ approaches 0, load type l and vehicle type v become less compatible. We use piecewise-linear value function approximations and update them by using (19) and (20) with $\alpha_n = 20/(40+n)$.

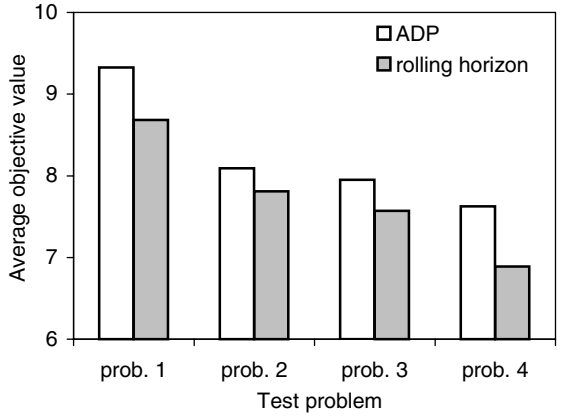
We begin by exploring the performance of ADP on problems where $\{(\widehat{R}_t, \widehat{D}_t)\colon t \in \mathcal{T}\}$ are deterministic. These problems can be formulated as integer multicommodity min-cost network flow problems as in problem (29); we solve their linear programming relaxations to obtain upper bounds on the optimal objective values. Table 3 shows the ratios of the objective values obtained by ADP and by the linear programming relaxations. ADP obtains objective values within 3% of the upper bounds on the optimal objective values.

We use the so-called rolling horizon strategy as a benchmark for problems where $\{(\widehat{R}_t, \widehat{D}_t)\colon t \in \mathcal{T}\}$ are random. The N-period rolling horizon strategy solves an integer multicommodity min-cost network flow problem to make the decisions at time period t. This problem is similar to problem (29), but it "spans" only the time periods $\{t, t+1, \ldots, t+N\}$, as opposed to "spanning" the time periods $\{0, \ldots, T\}$. The first time period t in this problem involves the known realization of $(\widehat{R}_t, \widehat{D}_t)$, and the future time periods $\{t+1, \ldots, t+N\}$ involve the expected values of $\{(\widehat{R}_{t+1}, \widehat{D}_{t+1}), \ldots, (\widehat{R}_{t+N}, \widehat{D}_{t+N})\}$. After solving this problem, we only implement the decisions for time period t and solve a similar problem when making the decisions for time period $t+1$. Figure 7 shows the average objective values obtained in the testing iterations, where the white and the gray bars, respectively, correspond to ADP and the rolling horizon strategy. The results indicate that ADP performs noticeably better than the rolling horizon strategy.

8. Extensions and Final Remarks

In this chapter, we described a modeling framework for large-scale resource allocation problems, along with a fairly flexible algorithmic framework that can be used to obtain good

FIGURE 7. Performances of ADP and the rolling horizon strategy on different test problems.



solutions for them. There are still important questions—some of which have already been addressed by the current research and some of which have not—that remain unanswered in this chapter.

Our modeling framework does not put a restriction on the number of dimensions that we can include in the attribute space. On the other hand, our algorithmic framework uses value function approximations of the form $\overline{V}_t^x(R_t^x) = \sum_{a \in \mathcal{A}} \overline{V}_{ta}^x(R_{ta}^x)$, which implicitly assumes one can enumerate all elements of $\mathcal{A}$. This issue is not as serious as the curse of dimensionality mentioned in §3, which is related to the number of possible values that the state vector S_t can take, but it can still be a problem. For example, considering the attribute vector in (2) and assuming that there are 100 locations in the transportation network, 10 possible values for the travel time, 8 possible values for the time on duty, 5 possible values for the number of days away from home, and 10 possible vehicle types, we obtain an attribute space that includes 40,000,000 ($= 100 \times 10 \times 8 \times 5 \times 10 \times 100$) attribute vectors. In this case, because problem (13) includes at least $|\mathcal{A}|$ constraints, solving this problem would be difficult. We may use the following strategy to deal with this complication. Although $\mathcal{A}$ may include many elements, the number of available resources is usually small. For example, we have several thousand vehicles in the fleet-management setting. In this case, we can solve problem (13) by including only a subset of constraints (4) whose right side satisfies $R_{ta} + \widehat{R}_{ta} > 0$. This trick reduces the size of these problems. However, after such a reduction, we are not able to compute ϑ_{ta}^n for all $a \in \mathcal{A}$. This difficulty can be remedied by resorting to aggregation strategies; we can approximate ϑ_{ta}^n in (17) by using $\vartheta_{ta'}^n$ for some other attribute vector a' such that a' is "similar" to a and $R_{ta'} + \widehat{R}_{ta'} > 0$.

Throughout this chapter, we assumed that there is a single type of resource and all attribute vectors take values in the same attribute space. As mentioned in §2, we can include multiple types of resources in our modeling framework by using multiple attribute spaces, say $\mathcal{A}^1, \ldots, \mathcal{A}^N$, and the attribute vectors for different types of resources take values in different attribute spaces. Unfortunately, it is not clear how we can construct good value function approximations when there are multiple types of resources. Research shows that straightforward separable value function approximations of the form $\overline{V}_t^x(R_t^x) = \sum_{n=1}^N \sum_{a \in \mathcal{A}^n} \overline{V}_{ta}^x(R_{ta}^x)$ do not perform well.

Another complication that frequently arises is the advance information about the realizations of future random variables. For example, it is common that shippers call in advance for future loads in the fleet-management setting. The conventional approach in Markov decision processes to address advance information is to include this information in the state vector. This approach increases the number of dimensions of the state vector, and it is not clear how to approximate the value function when the state vector includes such an extra dimension.

We may face other complications depending on the problem setting. To name a few for the fleet-management setting, the travel times are often highly variable, and using expected values of the travel times does not yield satisfactory results. The load pickup windows are almost always flexible; we have to decide not only which loads to cover but also when to cover these loads. The decision-making structure is often decentralized, in the sense that the decisions for the vehicles located at different locations are made by different dispatchers.

9. Bibliographic Remarks

The approximate dynamic programming framework described in this chapter has its roots in stochastic programming, stochastic approximation, and dynamic programming. Birge and Louveaux [3], Ermoliev and Wets [11], Kall and Wallace [16], Kushner and Clark [18], and Ruszczynski and Shapiro [27] provide thorough introductions to stochastic programming and stochastic approximation. Puterman [25] covers the classical dynamic programming theory, whereas Bertsekas and Tsitsiklis [2] and Sutton and Barto [31] cover the approximate dynamic programming methods more akin to the approach followed in this chapter.

The modeling framework in §2 is a simplified version of the one described in Powell et al. [23]. Shapiro [28] develops a software architecture that maps this modeling framework to software objects. Powell et al. [24] uses this modeling framework for a driver scheduling problem.

The approximate dynamic programming framework in §3 captures the essence of a long line of research documented in Godfrey and Powell [13, 14], Papadaki and Powell [19], Powell and Carvalho [20, 21], and Topaloglu and Powell [35]. The idea of using simulated trajectories of the system and updating the value function approximations through stochastic approximation-based methods bears close resemblance to temporal differences and Q-learning, which are treated in detail in Sutton [30], Tsitsiklis [36], and Watkins and Dayan [41]. Numerous methods have been proposed to choose a good set of values for the adjustable parameters in the generic value function approximation structure in (15). Bertsekas and Tsitsiklis [2] and Tsitsiklis and Van Roy [37] propose simulation-based methods, Adelman [1] and de Farias and Van Roy [10] utilize the linear programming formulation of the dynamic program, and Tsitsiklis and Van Roy [38] uses regression.

Birge and Wallace [4] and Wallace [40] use piecewise-linear functions to construct bounds on the value functions arising from multistage stochastic programs, whereas Cheung and Powell [6, 7] use piecewise-linear functions to construct approximations to the value functions. The approaches used in these papers are static; they consider all possible realizations of the random variables simultaneously rather than using simulated trajectories of the system to iteratively improve the value function approximations.

In §4, the idea of using linear value function approximations is based on Powell and Carvalho [21]. Godfrey and Powell [12] proposes a method, called concave adaptive value estimation, to update piecewise-linear value function approximations. This method also uses a "local" update of the form (19). The methods described in §4 to update piecewise-linear value function approximations are based on Kunnumkal and Topaloglu [17], Powell et al. [22], and Topaloglu and Powell [33].

Scenario-based stochastic programming methods described in §6 date back to Dantzig and Ferguson [9]. Wets [42, 43] treat these methods in detail. There are several variants of Benders decomposition-based methods; L-shaped decomposition method, stochastic decomposition method, and cutting plane and partial sampling method are three of these. L-shaped decomposition method is due to Van Slyke and Wets [39], stochastic decomposition method is due to Higle and Sen [15], and cutting plane and partial sampling method is due to Chen and Powell [5]. Ruszczynski [26] gives a comprehensive treatment of these methods. Stochastic hybrid approximation procedure is due to Cheung and Powell [8].

Some of the computational results presented in §7 are taken from Topaloglu and Powell [35].

There is some research that partially answers the questions posed in §8. Powell et al. [24] uses the aggregation idea to solve a large-scale driver scheduling problem. Spivey and Powell [29] systematically investigates different aggregation strategies. Topaloglu [32] and Topaloglu and Powell [34] propose value function approximation strategies that allow decentralized decision-making structures. Topaloglu [32] presents a method to address random travel times.

References

[1] D. Adelman. A price-directed approach to stochastic inventory routing. *Operations Research* 52(4):499–514, 2004.

[2] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming.* Athena Scientific, Belmont, MA, 1996.

[3] J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming.* Springer-Verlag, New York, 1997.

[4] J. R. Birge and S. W. Wallace. A separable piecewise linear upper bound for stochastic linear programs. *SIAM Journal of Control and Optimization* 26(3):1–14, 1988.

[5] Z.-L. Chen and W. B. Powell. A convergent cutting-plane and partial-sampling algorithm for multistage linear programs with recourse. *Journal of Optimization Theory and Applications* 103(3):497–524, 1999.

[6] R. K. Cheung and W. B. Powell. An algorithm for multistage dynamic networks with random arc capacities, with an application to dynamic fleet management. *Operations Research* 44(6):951–963, 1996.

[7] R. K.-M. Cheung and W. B. Powell. Models and algorithms for distribution problems with uncertain demands. *Transportation Science* 30(1):43–59, 1996.

[8] R. K.-M. Cheung and W. B. Powell. SHAPE-A stochastic hybrid approximation procedure for two-stage stochastic programs. *Operations Research* 48(1):73–79, 2000.

[9] G. Dantzig and A. Ferguson. The allocation of aircrafts to routes: An example of linear programming under uncertain demand. *Management Science* 3:45–73, 1956.

[10] D. P. de Farias and B. Van Roy. The linear programming approach to approximate dynamic programming. *Operations Research* 51(6):850–865, 2003.

[11] Y. Ermoliev and R. J.-B. Wets, editors. *Numerical Techniques for Stochastic Optimization.* Springer-Verlag, New York, 1988.

[12] G. A. Godfrey and W. B. Powell. An adaptive, distribution-free approximation for the newsvendor problem with censored demands, with applications to inventory and distribution problems. *Management Science* 47(8):1101–1112, 2001.

[13] G. A. Godfrey and W. B. Powell. An adaptive, dynamic programming algorithm for stochastic resource allocation problems I: Single period travel times. *Transportation Science* 36(1):21–39, 2002.

[14] G. A. Godfrey and W. B. Powell. An adaptive, dynamic programming algorithm for stochastic resource allocation problems II: Multi-period travel times. *Transportation Science* 36(1):40–54, 2002.

[15] J. L. Higle and S. Sen. Stochastic decomposition: An algorithm for two stage linear programs with recourse. *Mathematics of Operations Research* 16(3):650–669, 1991.

[16] P. Kall and S. W. Wallace. *Stochastic Programming.* John Wiley and Sons, New York, 1994.

[17] S. Kunnumkal and H. Topaloglu. Stochastic approximation algorithms and max-norm "projections." Technical report, Cornell University, School of Operations Research and Industrial Engineering, Ithaca, NY, 2005.

[18] H. J. Kushner and D. S. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems.* Springer-Verlag, Berlin, Germany, 1978.

[19] K. Papadaki and W. B. Powell. An adaptive dynamic programming algorithm for a stochastic multiproduct batch dispatch problem. *Naval Research Logistics* 50(7):742–769, 2003.

[20] W. B. Powell and T. A. Carvalho. Dynamic control of multicommodity fleet management problems. *European Journal of Operations Research* 98:522–541, 1997.

[21] W. B. Powell and T. A. Carvalho. Dynamic control of logistics queueing network for large-scale fleet management. *Transportation Science* 32(2):90–109, 1998.

[22] W. B. Powell, A. Ruszczynski, and H. Topaloglu. Learning algorithms for separable approximations of stochastic optimization problems. *Mathematics of Operations Research* 29(4):814–836, 2004.

[23] W. B. Powell, J. A. Shapiro, and H. P. Simão. A representational paradigm for dynamic resource transformation problems. C. Coullard, R. Fourer, and J. H. Owens, eds. *Annals of Operations Research.* J. C. Baltzer AG, 231–279, 2001.

[24] W. B. Powell, J. A. Shapiro, and H. P. Simão. An adaptive dynamic programming algorithm for the heterogeneous resource allocation problem. *Transportation Science* 36(2):231–249, 2002.

[25] M. L. Puterman. *Markov Decision Processes.* John Wiley and Sons, New York, 1994.

[26] A. Ruszczynski. Decomposition methods. A. Ruszczynski and A. Shapiro, eds., *Handbook in Operations Research and Management Science*, Volume on *Stochastic Programming.* North-Holland, Amsterdam, The Netherlands, 2003.

[27] A. Ruszczynski and A. Shapiro, editors. *Handbook in Operations Research and Management Science*, Volume on *Stochastic Programming.* North-Holland, Amsterdam, The Netherlands, 2003.

[28] J. A. Shapiro. A framework for representing and solving dynamic resource transformation problems. Ph.D. thesis, Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ, 1999.

[29] M. Z. Spivey and W. B. Powell. The dynamic assignment problem. *Transportation Science* 38(4):399–419, 2004.

[30] R. S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning* 3:9–44, 1988.

[31] R. S. Sutton and A. G. Barto. *Reinforcement Learning*. The MIT Press, Cambridge, MA, 1998.

[32] H. Topaloglu. A parallelizable dynamic fleet management model with random travel times. *European Journal of Operational Research*. Forthcoming.

[33] H. Topaloglu and W. B. Powell. An algorithm for approximating piecewise linear functions from sample gradients. *Operations Research Letters* 31:66–76, 2003.

[34] H. Topaloglu and W. B. Powell. A distributed decision making structure for dynamic resource allocation using nonlinear functional approximations. *Operations Research* 53(2):281–297, 2005.

[35] H. Topaloglu and W. B. Powell. Dynamic programming approximations for stochastic, time-staged integer multicommodity flow problems. *INFORMS Journal on Computing* 18(1):31–42, 2006.

[36] J. N. Tsitsiklis. Asynchronous stochastic approximation and Q-learning. *Machine Learning* 16:185–202, 1994.

[37] J. Tsitsiklis and B. Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control* 42:674–690, 1997.

[38] J. Tsitsiklis and B. Van Roy. Regression methods for pricing complex American-style options. *IEEE Transactions on Neural Networks* 12(4):694–703, 2001.

[39] R. Van Slyke and R. Wets. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal of Applied Mathematics* 17(4):638–663, 1969.

[40] S. W. Wallace. A piecewise linear upper bound on the network recourse function. *Mathematical Programming* 38:133–146, 1987.

[41] C. J. C. H. Watkins and P. Dayan. Q-learning. *Machine Learning* 8:279–292, 1992.

[42] R. Wets. Programming under uncertainty: The equivalent convex program. *SIAM Journal of Applied Mathematics* 14:89–105, 1966.

[43] R. J.-B. Wets. Stochastic programs with fixed recourse: The equivalent deterministic problem. *SIAM Review* 16:309–339, 1974.

TUTORIALS IN
OPERATIONS RESEARCH
INFORMS 2006

 | ISBN 13 978-1-877640-20-9
DOI 10.1287/educ.1063.0028

Enhance Your Own Research Productivity Using Spreadsheets

Janet M. Wagner and Jeffrey Keisler
Department of Management Science and Information Systems, University of Massachusetts, Boston, 100 Morrissey Boulevard, Boston, Massachusetts 02125
{janet.wagner@umb.edu, jeff.keisler@umb.edu}

Abstract Spreadsheets are the modeling tool of choice for many OR/MS researchers. Surveys of users show that most do not use basic good practices, and most large spreadsheets have flaws leading to results ranging from wasted time to downright scandal. Fortunately, many solutions to these problems are already known and easily learned. This workshop, taught by OR/MS modelers who have firsthand experience with both "sin" and "salvation" in the spreadsheet kingdom, presents the authors' "top 10" Excel methods and 4 major spreadsheet applications from their own research and professional lives. Tutorial participants, bring your laptops!

Keywords productivity; spreadsheet modeling; information systems; spreadsheets

1. Introduction

Like Rodney Dangerfield, spreadsheets don't get no respect. Casimer [5] proclaimed "Real Programmers Don't Use Spreadsheets." Grossman et al. [11] describe multiple examples showing a "perception that spreadsheets are somehow different than other programming tools, and that spreadsheets are suitable for personal use but not for important tasks which are reserved to information systems" (p. 2).

However, the use of spreadsheets is ubiquitous in both business and OR/MS. Microsoft Excel alone has an installed user base of 440 million licenses (Microsoft [15]), with additional hundreds of millions using Open Office, Quattro Pro, Lotus 123, and Gnumeric. Scaffidi et al. [22] estimates that the number of spreadsheet and database users in the United States alone will reach 55 million in 2012, over four times their estimate of 13 million "professional" programmers. Evidence is growing about the many uses of spreadsheets for critical business processes. For example, the paper "Stop That Subversive Spreadsheet" by Butler and Chadwick [4] describes the nexus of concerns of both academicians and practitioners that led to the formation of the European Spreadsheet Risk Interest Group (EuSPRIG) [10]. As just one example, Croll [7] talks about the ubiquitousness of spreadsheets in the London financial community (called the "City of London"), and concludes "it is completely within the realm of possibility that a single, large, complex but erroneous spreadsheet could directly cause the accidental loss of a corporation or institution, significantly damaging the City of London's reputation" (p. 91). Estimates of the number of OR/MS spreadsheet users are harder to come by. However, the extent of the coverage of spreadsheets in OR/MS textbooks and the existence of groups such as EuSPRIG and, within INFORMS, of the Spreadsheet Productivity Research Interest Group (SPRIG) [25] provide evidence that spreadsheets are a common tool for those in OR/MS fields.

The focus of this tutorial is specifically on the use of spreadsheets as OR/MS application development tools. The goal of this tutorial is not just to develop spreadsheet examples similar to those available in a comprehensive Excel manual, but rather to gain an understanding

at an abstract level of what spreadsheet tools are and how to relate them to specific OR/MS modeling needs. In this tutorial, we will provide concepts and methods for building, verifying, and using spreadsheets in a way that maximally enhances productivity. We will also present examples of spreadsheets, developed and used in the authors' professional lives, to both model good spreadsheet practice and to illustrate our concept of matching spreadsheet tools to real professional OR/MS needs.

2. Spreadsheets: From "Sin" to "Salvation"

Spreadsheets can be almost too easy to use. It is quite possible for OR/MS models to push spreadsheets to (and beyond?) the limits of their capabilities. Have you ever built a large, complex spreadsheet model that ended up taking you more time to debug than the original development time? When you revise an article after six months, do you have to spend large amounts of time remembering exactly how your spreadsheet works? Is there a significant chance your model is actually invalid?

EuSPRIG [10] maintains press accounts of important spreadsheet mistakes on its website; there were 85 such stories when this tutorial was written. Recent examples include the City Council of Las Vegas having to postpone their vote on the city budget because of over five million dollars of errors in the spreadsheet output provided as part of the budget bill, and several examples of companies having to restate earnings by millions of dollars due to "clerical errors" in spreadsheets. Striking in this archive is the magnitude of the effects of the reported mistakes and the fact that, despite the magnitude and criticality of these applications, the mistakes occur mainly from simple common mistakes such as botched sorting or misspecified sum ranges. We would all like to keep ourselves and our spreadsheet exploits out of the EuSPRIG error archive (and the press), but, undoubtedly, so did the authors and users of those reported incidents.

The challenge, then, is that we are all "sinners" regarding robust and rigorous spreadsheet design and implementation. In this tutorial, we will explore the path of "salvation," paying specific attention to certain paving stones along that path. We believe that, like any other information system application, spreadsheets pose risks. However, many straightforward techniques exist that can help reduce and manage those risks. The opportunities spreadsheets provide are simply too numerous to dismiss this technology completely, even when developing complex systems.

3. Sources of Salvation (Background Knowledge)

Strategies for the effective and efficient use of spreadsheets can be drawn from a number of areas, including software development and engineering, OR/MS modeling, the psychology of error, and traditional auditing. In addition, commercial applications to assist with spreadsheet development and use appear on the market almost daily. We will give some selected representative sources for these background areas below. We also recommend both the EuSPRIG [10] and SPRIG [25] websites, which maintain links to a variety of research articles, conference presentations, books, and products related to spreadsheet modeling and development.

Software development and engineering: Current spreadsheet practice has been compared to the "Wild West" days of early programmers. The disciplines and methods of the field of software engineering, which have helped to tame the development of conventional software, have much to offer spreadsheet developers as well. Boehm and Basili [3] provide data that show "disciplined personal practice can reduce deficit introduction rates [in programs] up to 75%" (p. 136). Textbooks and reference works on software engineering include those by McConnell [13, 14], Pressman [19], and Sommerville [24].

OR/MS modeling: Spreadsheet applications of OR/MS models and techniques have become an integral part of many textbooks and reference books. Multiple examples can probably be best obtained in the exhibit halls accompanying this conference, but "classics"

would include books by Albright and Winston [1], Powell and Baker [18], Ragsdale [20], and Serif et al. [23]. Tennent and Friend [27] is another useful book, written for economists.

Psychology of error: Humans make errors, and psychologists, among others, have studied factors that can lead to either more or less of them. Ray Panko maintains a Web page [26] with a comprehensive bibliography on both human error in general and spreadsheet errors in particular.

Traditional auditing: The process of reviewing the accuracy of financial statements has much in common with processes for reviewing the accuracy of spreadsheets. Basic textbooks on auditing include those by Arens et al. [2] and Rittenberg and Schwieger [21]. The previously mentioned SPRIG website [25] contains a listing of available packages for spreadsheet auditing. O'Beirne [17] is a useful spreadsheet-oriented book, covering auditing as well as spreadsheet design topics.

4. Process and Principles for Salvation (Spreadsheet Design and Engineering)

Paradoxically, research productivity using spreadsheets is probably most enhanced by *investing* time—as long as that time is spent before touching a keyboard. Following Powell and Baker [18] we advocate following a thoughtful process for spreadsheet development, with separate phases of spreadsheet design, building, and testing. As Powell and Baker point out, builders do not build buildings without blueprints and neither should researchers build spreadsheets without plans.

Principles adapted from Powell and Baker for ease of use and for avoiding the dreaded "spaghetti code" include the following:

- Separating data from calculations and separating analysis from presentation;
- Organizing spreadsheets with a logical progression of calculations (top to bottom, left to right);
- Developing data and analytical "modules" (including grouping within worksheet, and the worksheet structure itself);
- Sketching, in advance of development, major spreadsheet elements and calculation flow;
- Using graphical aids to modeling (we are particular fans of influence diagrams);
- Giving thought to and consulting with the end users of the spreadsheet on their needs (the user, who is not necessarily the spreadsheet builder, may have a very different view of the process than the spreadsheet analyst);
- Keeping formulas short and simple;
- Planning for documentation "as you go;"
- Stating model assumptions explicitly;
- Using formatting aids, such as color, text differences, cell outlining; and
- Protecting end-users from unnecessary analytical details and inadvertent changes.

In Excel, basic built-in tools supporting these principles include the following:

- Availability of absolute versus relative references;
- Cell and text formatting;
- Protected and locked worksheets and cells;
- Data (range) names; and
- Function wizards.

We assume readers are familiar with these basic tools, although we will quickly go over them as requested in a "hands-on" manner in the tutorial session. Readers unfamiliar with these Excel elements can explore their use using the built-in help, a basic Excel text (Harvey [12]), or in Powell and Baker [18]. (Or, of course, using the time-honored approach of asking a friend.)

We also suggest that investing time exploring these basic tools, before any research or modeling efforts, is likely to pay multiple dividends. Both of us have systematically examined

all the available functions and cell and text formatting options in Excel. We found this investment of time exploring spreadsheet capabilities is repaid many times over by the new ideas and possibilities for their application that we gain from it. Walkenbach's [28] *Excel 2003 Bible* is a comprehensive Excel book, favored by the authors.

5. On the Path to Salvation (Advanced Tools)

More advanced (and lesser known) Excel tools are available that, if properly and consistently used, can aid in the efficient, effective development and use of research and end-user spreadsheets. In this section, we will give some "step-by-step" directions as well as hints on the use of the following Excel methods:

- Comment and formula display options;
- Data validation;
- Spreadsheet auditing; and
- Built-in error checking.

Note: Material in italic describes MS Excel (Office 2003) commands.

Comment and formula display options: A text comment to accompany a cell is added by the following.

Insert-Comment. Comments do not have to clutter up the spreadsheet, because the default is to show them only when the cursor is on the particular cell. (A cell with comments is indicated by a red triangle in the corner of the commented cell.) Comments are a good way to document calculations so a given formula is understandable six months from now.

Tools-Option-View gives different options. A comment can be removed by *Edit-Clear-Comments*.

To see a formula and a color-coded display of the cells referenced in the formula, double click on the cell, or use F2. All formulas in a worksheet can be displayed simultaneously by pressing *Ctrl* + ˜ (tilde).

Data validation: If users enter data into a spreadsheet, guidance can be provided to them (and errors avoided) by using: *Data-Validation*. When data validation is required for a cell, the value can be restricted (e.g., "between 0 and 50") as can the type of value (e.g., "whole number"). Data validation menu items also allow comments to be specified that will show when the cell is selected as well as the error message that will appear when the data is not entered according to the specifications. *Data-Validation-Clear All* removes the validation specifications.

Spreadsheet auditing: Excel comes with built-in formula-auditing functions, which are accessed by *Tools-Formula Auditing-Show Audit Toolbar*. These auditing functions are particularly helpful in parsing and testing complex formulas. The audit toolbar has tools that graphically trace the cells used in a formula (*Trace References*), or trace where a particular cell is used in a subsequent formula (*Trace Dependents*). Another useful function in the audit toolbar is *Evaluate Formula* that shows steps in a complex formula calculated a piece at a time.

Error checking: Starting in Excel 2002, Excel looks for certain basic errors in formulas. We note that, like spell and grammar check in word processing programs, some people find these checks more annoying than helpful. *Tools-Options-Error Checking* brings up a menu that allows adjustment for what errors are and are not looked for (and/or turn error checking completely on or off, as wished).

All the above general purpose tools will enhance the development process for all spreadsheets. We maintain that due to the complexity of most OR/MS models, building on a solid platform of good spreadsheet practices is particularly important. Models with frequent comments for complex formulas, which have had their formulas audited, have been error checked, and with built-in data validation they, most likely, will be able to be followed six months from now, can be turned over to a successor with ease, and will be easier to test and use.

6. The End of the Road: Putting It All Together (Techniques and Applications)

The focus of this tutorial is to find a mathematically elegant way to use the structure and functionality available in spreadsheets to encode the structure of your problem. In this section, we will go over our "top 10" set of Excel methods for OR/MS researchers. We will motivate this list by showing examples of how we have combined these specific "top 10" tools, and the more general good spreadsheet design principles discussed in previous sections, into "killer aps."

We start by observing that Excel methods can be classified broadly as "interface" tools and "analysis tools." Most applications will require both types of tools, but the balance of these two functions will vary by the application and intended use. A spreadsheet intended to answer a research question may focus mainly on the analytical tools with little attention to interface/presentation, while another system intended to support nontechnical decision makers may require mainly interface tools. Careful consideration, however, needs to be given to both functions—no matter the application.

6.1. Interface Tools

6.1.1. How We Doin'? A Spreadsheet for Nationally Normed Student Survey Results.

This application came from one author's foray into college administration, where an OR/MS sensibility infused (for good or for ill) the position of Associate Dean. The value of this spreadsheet application is in its ability to present large amounts of data in a compact and engaging form. The file is available as studentsurvey.xls.* Note that the data in this spreadsheet has been altered, both for UMass Boston and the benchmarking information. The values in this example are representative values, not the actual ones.

The College of Management at UMass Boston, like many AACSB-accredited schools, participates in student assessment surveys using an instrument from Educations Benchmarking Inc. (EBI) [9]. EBI surveys have the advantage not only of providing the responses of our own students, but providing national benchmarks as well (and comparison data for six other benchmark institutions). EBI provides multiple analyses and presentations of results, but we found it difficult to both interpret and distribute the results of these surveys. The spreadsheet presented here provides an interactive graphical representation for each of the 66 survey questions, showing in one compact, user-friendly display UMass Boston's results, compared to the six benchmark schools, the set of schools in our same Carnegie classification, and the entire set of schools using EBI that particular year (see Figure 1).

This first example relied heavily on the interface focused tools of

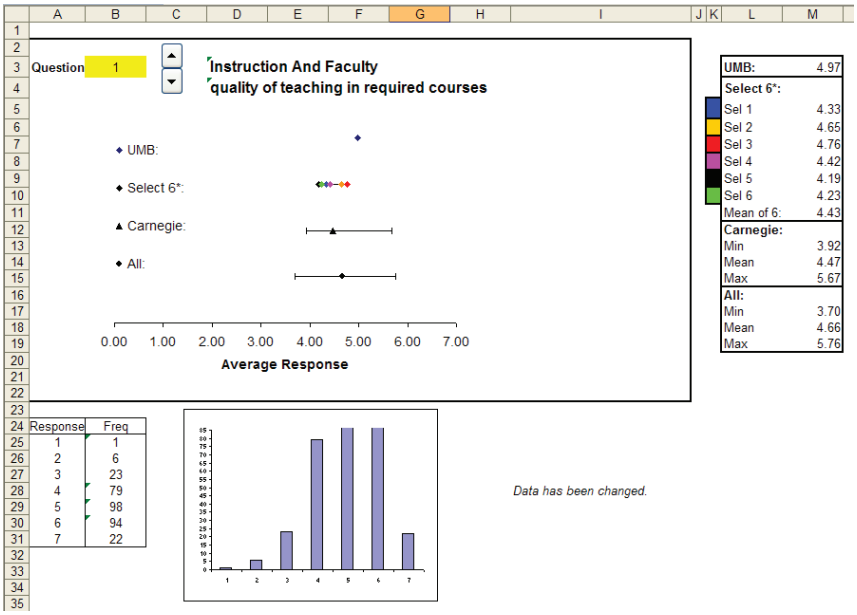
(1) conditional formatting,
(2) graphs, and
(3) form controls.

Method 1: Conditional formatting. Excel allows the user to change the formatting of what is displayed in a cell, depending on the value (or formula) in the cell, a feature accessed by *Format-Conditional Formatting.* The resulting menu allows the user to set one (or more) conditions that will then lead to a specified format (including cell background, font, and cell outlining).

Method 2: Graphs. The ability to simultaneously present information in text, numbers, and graphs is one of the major sources of the power of spreadsheets. The graphical features of Excel can be accessed by *Insert-Chart* (or by clicking on the chart icon in the standard toolbar). This tutorial does not include a comprehensive discussion of all the Excel graph options. However, we want to highlight that particularly interesting interfaces can be created by using "live" graphs, which respond to changes in user input data.

*All spreadsheets referenced but not provided in this chapter are available on the *TutORials* website at http://tutorials.pubs.informs.org and on the SPRIG website at http://sprig.section.informs.org.

FIGURE 1. Student survey display.



It should be noted that there are also significant limitations to Excel graphs, particularly for more scientific and other technical uses. Multiple graphing computer packages exist, which should certainly be considered for specialized uses.

Method 3: Form controls. A set of interesting Excel controls can be revealed by *View-Toolbar-Forms*. We will focus on the two tools of *Scroll Bar* and the *Spinner*. Both controls are accessed by clicking on the appropriate icon, dragging an appropriately sized area on the spreadsheet itself, right-clicking on the control, and then choosing *Format Control*. These tools allow a "kinesthetic" way to enter or change data, and can be particularly useful in creating applications designed to engage the spreadsheet user in data manipulation. (We are not sure why moving a bar or clicking a little triangle is so much more engaging that retyping a number, but from experience we know that these tools really do draw users in.)

We encourage the reader to open and explore the first spreadsheet (studentsurvey.xls) here. Tools used to produce this spreadsheet include lookup functions (discussed below) and the creative use of formatting, form controls, and graphical functions of Excel. The graph is based on an x-y plot, with three separate data series. Note that some data are hidden (white text, in columns N through Q). The horizontal lines on the plots come from the error bar capability. The spinner is used to pick a question, which looks up the data for that question (both from the internal survey results and the national norms), and the graph then automatically redisplays.

This spreadsheet was given to all the college's standing committees, which included most of the tenure-track faculty. Discussions that semester, involving multiple staff and faculty groups, provided ample evidence that this spreadsheet was used by multiple people. The information gained from this effort resulted in several changes to existing procedures and new initiatives. At least partly as a result of these programmatic changes, when another survey was taken two years later, the undergraduate results improved on 63 out of the 66 questions.

6.2. Analytical Tools

6.2.1. Classrooms Need Chalk and Students: What Class Schedules Can Tell You. The second example is also a simplified version of a "real" spreadsheet, again, used for college administration. The file is available as classsched.xls. Again, this spreadsheet contains

representative data, not any actual semester's schedule. This spreadsheet involves some important but fairly simple calculations; however, its real value is its ability to present data in usable form. It started as a single-purpose spreadsheet, to calculate faculty deployment ratios (e.g., percent of MBA credits presented by full-time faculty) required by AACSB using basic information supplied by the registrar's class schedule and the college's faculty list. However, once this data set existed, questions that had never been imagined were posed about these data. Hence, this spreadsheet developed over several years, with a new report being created each time someone else said, "could you tell me ...?" In this case, the presentation issue is that data available from the run-of-the-mill class schedule has multiple uses and needs to be displayed in multiple ways.

The second example is based on the analytically focused tools of

(4) lookup functions,
(5) sorting,
(6) filtering, and
(7) pivot table.

Method 4: Lookup functions. The lookup and reference functions are some of the most useful Excel functions in creating high-functioning spreadsheet systems. We will focus on the HLOOKUP and VLOOKUP functions, but all of the lookup and reference functions are worth a look. These functions can be accessed by: *Insert-Function* (or from the *fx* icon). The HLOOKUP function is used to look up a value across a row; the VLOOKUP function is used when you are looking for a value down a column. Among other uses, these functions can be used to obtain functionality similar to a relational database. They can also enable data to be entered in a logical and compact form, so that entries can be built up from components instead of having to retype data multiple times. For example, to compile a list of faculty members, one can use a LOOKUP function to determine what college a given department is in instead of having to remember and type it each time.

Method 5: Sorting. Before we discuss this method, we need to point out that sorting is a double-edged sword. The ability to sort information, by rows or by columns, is both one of the most useful (and used) Excel capabilities *and* is also a way to cause really serious errors. Sorting capabilities are accessed by selecting the range containing the data to be sorted then *Data-Sort*. Where errors commonly occur is in selecting the incorrect range of data to be sorted. Sorting should be done with care. If one was to sort all but one column of a given table, the error can only be corrected using the "undo" function, which means if the error is not caught quickly, it may not be fixable at all. Using named ranges for data that are to be frequently sorted is a good way to reduce the occurrence of such errors.

Method 6: Filtering and subtotals. Filtering allows the user to choose a subset of a data range, according to a user-defined criteria, for data organized in columns with column headings. Filtering is accessed by selecting a column label (or labels) and then *Data-Filter-AutoFilter*. Small triangles then appear at the top of the columns. Selecting the triangle shows a list of values in the column; clicking on a value filters for that value. More advanced custom filters can be created with other menu options. The triangles can be turned off (and the full unfiltered set of data restored) by repeating *Data-Filter-AutoFilter*.

Helpful to use with filtering is the SUBTOTAL function, which we find useful if rather nonintuitive. Subtotal has two arguments, the first is a number that defines the calculation (use nine to get a sum), and the second is the data range to be used in the calculation. When no filter is applied, SUBTOTAL works like whatever function the user chooses (so with nine, Excel would calculate a regular sum). However, when the data is filtered, SUBTOTAL only calculates the chosen function for the displayed value (e.g., shows a subtotal).

Method 7: Pivot table. In a way, pivot tables are an extension of the subtotal function. For example, suppose a user had a list of employees, with associated departments and salaries. One could manually construct a table of total salary budget by department by using the filter and the SUBTOTAL function to choose each department in turn, and then recording

that department's total salary. The pivot table function, however, will create this table automatically.

A pivot table works only on data arranged in columns with a column label entered for every column. The pivot table is accessed by *Data-PivotTable and PivotChart Report*. The first two menus are fairly self-explanatory; at the third, click on *Layout*. Here, one has a chance to set up a table. The data are broken down by variables that are dragged to the row or column area. (So, in the departmental salary example, the department would be put in the column space.) The values to be broken down (salaries in the example) are dragged into the data area, and by clicking on the label in the data area, the calculations to be performed can be changed. To filter what values get into the pivot table, other variables can be put into the page area. Click *OK* then *finish*, and the breakdown (or pivot) table will appear.

Pivot tables are a very rich resource, and there is more to them than can be explained in this short tutorial. Chapter 21 of Walkenbach [28] discusses pivot tables in more detail. We have found that pivot tables are another example of a function that once a user grasps the basic idea, much of the rest can be picked up by playing around with them.

We encourage the reader to open and explore the second spreadsheet (classsched.xls) here. The spreadsheet for this second example was designed using the "good spreadsheet practice" of providing compact, logically organized data, followed by (separate) analyses, followed by (separate) presentations of the results. After the first worksheet, which provides a "front page" to the spreadsheet (see Figure 2), the next three worksheets are data (course list, instructor list, and then class sections). Filtering, sorting (macros attached to buttons using simple VBA code), and lookup functions help keep the data compact and organized, and reduce errors by drastically reducing retyping (and allowing quick, reliable data changes). The next worksheet (see Figure 3) includes the pivot tables necessary for the ratio analysis. Because these pivot tables are used only by the analyst, no particular attempt was made to make them user friendly. The following sheets focus more on presentation, covering a wide range of uses and presentations. As well as a managerial presentation of the ratio results, reports exist to show scheduling (which nights MBA classes are offered, see Figure 4), faculty workload (number of courses and total students, see Figure 5), a more user-friendly presentation of the class schedule, and a report to ensure that nobody is double scheduled (which, from sad experience, turned out to be important to check).

This system for semester class scheduling has been used for more than five years. It is used prospectively (as the semester schedule is being determined) and retrospectively

FIGURE 2. Class schedule front page.

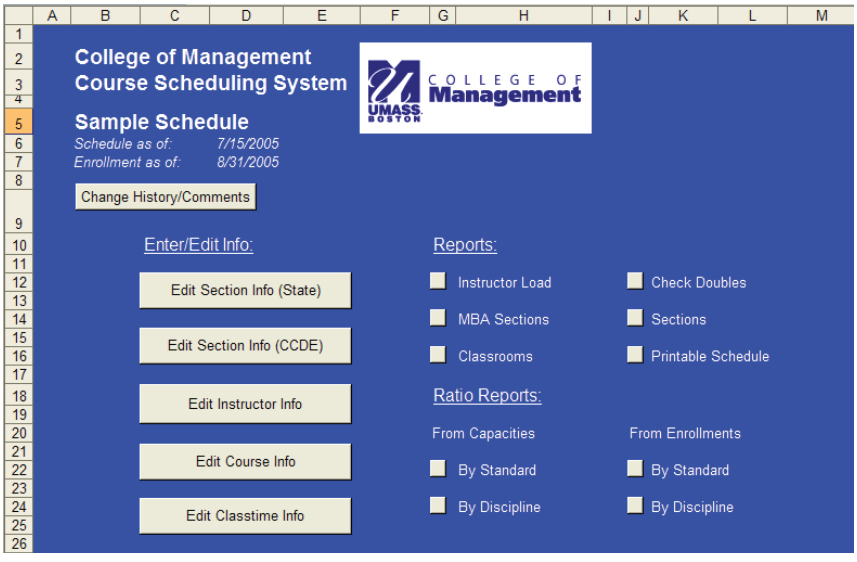


FIGURE 3. Class schedule pivot tables.

	A	B	C	D	E	F	G	H	I
1									
2	Return to Front								
3									
4	FT	(All)							
5									
6	Sum of SCH (Cap)	Discipline							
7	Program	ACCT	FIN	MGT	MIS	MKT	MS	ACM	Grand Total
8	UG Day	1968	1368	2166	1668	798	1572	636	10176
9	UG Eve	570	228	912	228	228	342	84	2592
10	Grad	735	735	1155	315	315	420		3675
11	Grand Total	3273	2331	4233	2211	1341	2334	720	16443
12									
13	FT	1							
14									
15	Sum of SCH (Cap)	Discipline							
16	Program	ACCT	FIN	MGT	MIS	MKT	MS	Grand Total	
17	UG Day	1968	1026	1710	1668	684	1572	8628	
18	UG Eve	456	114	684	114	114	342	1824	
19	Grad	735	525	1050	315	315	420	3360	
20	Grand Total	3159	1665	3444	2097	1113	2334	13812	
21									
22	FT & (AQ or PQ)	1							
23									
24	Sum of SCH (Cap)	Discipline							
25	Program	ACCT	FIN	MGT	MIS	MKT	MS	Grand Total	
26	UG Day	1968	1026	1710	1668	684	1572	8628	
27	UG Eve	456	114	684	114	114	342	1824	
28	Grad	735	525	1050	315	315	420	3360	
29	Grand Total	3159	1665	3444	2097	1113	2334	13812	

(to provide historical reports). The spreadsheets are available on the internal college servers, and are used by the college's administration (Associate Dean and MBA Director), as well as by the Department Chairs and the clerical staff. It is part of how the college does business. We believe that the widespread use of this system has occurred because each user can access (and manipulate) these data in exactly the way s/he likes and needs to interact with them.

6.2.2. Up and About: Calculation of Seasonal Indices on Top of a General Linear Trend. The third example may be most useful as a teaching example (one author remembers seeing a version of this example at a Teaching Management Science Workshop). It is also a good example of the functionality that occurs from the creative exploitation of the flexibility in spreadsheets. The file is available as seasonal.xls.

A common forecasting method involves developing a time-series model with a linear trend and seasonal indices. The example in the spreadsheet involves U.S. Commerce Data (Survey of Current Business) on quarterly general merchandise sales (in millions of dollars) from 1979 to 1989 (obtained from DASL [8]). An example such as this traditionally would be used in a class on business statistics or operations management.

FIGURE 4. Class schedule MBA schedule display.

	A	B	C	D	E	F	G
1							
2	Return to Front						
3	CourseType	Grad Core					
4							
5	Count of Section	Classtime					
6	Section	M 6:00	TU 6:00	W 6:00	TH 6:00	Sa 9:30	Grand Total
7	AF 601		1				1
8	AF 610	1					1
9	AF 620				1		1
10	MGT 650			1	1	1	3
11	MGT 660		1				1
12	MGT 689		1				1
13	MKT 670	1					1
14	MSIS 600	1					1
15	MSIS 630		1				1
16	MSIS 635			1			1
17	MSIS 640	1					1
18	Grand Total	4	4	2	2	1	13
19							
20							

FIGURE 5. Class schedule faculty workload.

	A	B	C	D	E	F
1						
2	Return to Front					
3						
4						
5						
6	Dept	AF		Dept	MM	
7						
8	Sum of SecEq			Sum of SecEq		
9	Name	Total		Name	Total	
10	Brian	3		Alton	1	
11	Charles	3		Coppers	2	
12	Chap	3		Dorn	2	
13	Core	3		Eve	2	
14	Fred	2		George	2	
15	Graham	2		Gund	3	
16	Holt	3		Hardy	2	
17	Hughes	3		Henry	1	
18	Jackson	1		Lars	2	
19	Johnson	2		Lind	3	
20	Jones	3		Lorne	1	
21	Overton	3		McGowen	1	
22	Ryan	3		Mercy	3	
23	Varga	3		Naby	2	
24	Vegas	4		Powell	1	
25	Worthen	2		Rorn	3	
26	Zabby	4		Rundi	4	
27	Grand Total	47		Samuels	2	
28				Wall	2	
29				Werth	4	
30				Wurnd	3	
31				Zaylor	4	
32				Grand Total	50	
33						

This example relies on the analytical focused tools (probably familiar to most OR/MS professionals) of

(8) statistical add-ins (e.g., regression) and

(9) solver.

Method 8: Statistical add-ins. Excel has a number of built-in statistical functions that can be accessed by *Tools-Data Analysis*. (Note, the data analysis pack is not always part of the standard installation procedure for Excel, and may have to be added in later.) Multiple statistical functions are available, and most have easy-to-follow menus. Note that Excel is not a special-purpose statistical package, and thus is not considered as robust as several commercially available statistical packages. Some of the more advanced functions have—at least in the past—had errors, for example, with the handling of missing data. (See Microsoft [16] for a report on Microsoft's responses to these issues.) Nonetheless, as part of a larger system, the ability to include statistical analysis with other types of calculations makes Excel the statistical package of choice.

Method 9: Solver. Again, it is beyond the scope of this short tutorial to go through all aspects of solver. Solver is also an Excel add-in, and can be accessed by *Tools-Solver*. The user must specify the cell containing the objective value (the target cell), the decision variables (the changing cells), and the constraints (added one by one). The option screen allows the user to choose the solution method (linear, types of nonlinear, etc.). Solver is thoroughly discussed in several OR/MS textbooks such as Albright and Winston [1], Ragsdale [20], and Serif et al. [23].

The first worksheet (see Figure 6) calculates seasonal indices using the two-step Seasonal Index Method (cf. Chase et al. [6], chap. 12). First, a linear regression is run on the original data and used to calculate a predicted value for each quarter. Then, the ratio of the actual data to the predicted amount is calculated, and these ratios are averaged for each individual

FIGURE 6. One-step linear/seasonal calculations.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Original Data				Traditional Seasonal Analysis											
2	TIME	Quarter	GMER		Linear Pred	Actual/Fore	Seasonal Forecast	Linear Only Error-Sq	Seasonal Correction Error-Sq							
3	1	1	30363		27053.6	1.122	21606.95	10952027.61	76668360.29			**Regression**				
4	2	2	25812		27534.2	0.937	26225.29	2965837.396	170804.5956			SUMMARY OUTPUT				
5	3	3	26026		28014.7	0.929	26055.47	3954952.076	868.5139376							
6	4	4	38081		28495.3	1.336	38797.07	91886572.66	512749.9539			*Regression Statistics*				
7	5	1	21909		28975.8	0.756	23142.14	49939620.41	1520646.274			Multiple R	0.599927			
8	6	2	26615		29456.3	0.904	28056.09	8073227.172	2076749.881			R Square	0.359912			
9	7	3	27039		29936.9	0.903	27843.22	8397754.566	646774.2059			Adjusted R S	0.3443			
10	8	4	40170		30417.4	1.321	41414.17	95112555.21	1547956.021			Standard Err	8144.391			
11	9	1	23641		30898.0	0.765	24677.34	52663742.15	1073993.989			Observations	43			
12	10	2	30477		31378.5	0.971	29886.9	812746.0867	348216.2859							
13	11	3	30065		31859.1	0.944	29630.97	3218686.33	188377.7809			ANOVA				
14	12	4	43805		32339.6	1.355	44031.27	131455048.1	51199.02126				*df*	*SS*	*MS*	*F*
15	13	1	24256		32820.2	0.739	26212.53	73344848.09	3828004.765			Regression	1	1.53E+09	1.529E+09	23.053711
16	14	2	30976		33300.7	0.930	31717.71	5404258.596	550133.1725			Residual	41	2.72E+09	66331111	
17	15	3	30579		33781.3	0.905	31418.73	10254415.22	705141.5706			Total	42	4.25E+09		
18	16	4	45471		34261.8	1.327	46648.38	125646231	1386212.379							
19	17	1	25891		34742.3	0.745	27747.72	78346263.96	3447411.739				*Coefficients*	*tandard Err*	*t Stat*	*P-value*
20	18	2	33050		35222.9	0.938	33548.52	4721442.04	248519.9924			Intercept	26573.07	2527.984	10.511564	3.325E-13
21	19	3	33385		35703.4	0.935	33206.48	5375133.448	31869.65657			TIME	480.5455	100.0839	4.801428	2.12E-05
22	20	4	50311		36184.0	1.390	49265.48	199572726.3	1093115.401							
23	21	1	29177		36664.5	0.796	29282.91	56063020.34	11217.48652			**Seasonal Averages**				
24	22	2	36855		37145.1	0.992	35379.33	84140.46998	2177613.864			(Actual/Forecast)				
25	23	3	35418		37625.6	0.941	34994.23	4873564.968	179579.8026			Quarter				
26	24	4	53126		38106.2	1.394	51882.58	225595573.3	1546089.3			1	0.799			
27	25	1	30325		38586.7	0.786	30818.1	68255788.2	243152.1243			2	0.952			
28	26	2	37977		39067.3	0.972	37210.13	1188648.52	588083.2624			3	0.930			
29	27	3	37028		39547.8	0.936	36781.98	6349377.124	60524.06834			4	1.362			
30	28	4	54369		40028.3	1.358	54499.68	205654457.7	17078.51883							
31	29	1	31594		40508.9	0.780	32353.3	79475227.15	576531.2032			**Mean Sqauare Error:**				
32	30	2	37617		40989.4	0.918	39040.94	11373307.06	2027611.666			Linear Only		8547.448		
33	31	3	36297		41470.0	0.875	38569.74	26759710.27	5165327.903			Seasonal Correction		2047.504		
34	32	4	52457		41950.5	1.250	57116.79	110386031.4	21713624.27							

quarter. These average ratios are then used as the seasonal indices, and the seasonalized prediction is then calculated as the predicted linear regression value multiplied by the seasonal index. The first worksheet uses the statistical add-in for regression.

However, the interesting observation is that because regression is, in fact, an optimization method (minimizing the total least squares error), this two-step procedure (regression then smoothing) can be done in one step, resulting in a lower total error than doing the two steps separately. In the example, Worksheet 2 (see Figure 7) redoes the seasonal index calculations, using the nonlinear optimization capabilities of solver to find *simultaneously* the coefficients of the linear model *and* the seasonal indices (with the constraint that the seasonal indices add up to the number of seasonal periods, four in this case). Here, the reduction in total error is not high, but it is nonetheless reduced.

The value in this example is to develop, in students as well as in researchers, the creativity (supported by the flexibility of spreadsheets) to view and manipulate problems using a variety of methods. Traditional regression analysis and optimization are not commonly combined in this way.

FIGURE 7. Two-step linear/seasonal calculations.

	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Original Data			One-Step Seasonal Analysis									
2	Quarter	GMER		Linear Pred	Actual/Fore	Seasonal Forecast	Linear Only Error-Sq	Seasonal Correction Error-Sq					
3	1	30363		26838.7	1.131	20857.91	12420731.99	90346674.77					
4	2	25812		27346.7	0.944	25888.36	2355295.795	5830.615051					
5	3	26026		27854.7	0.934	25707.49	3344145.454	101448.9124					
6	4	38081		28							**One Step Results**		
7	1	21909		28								*Coefficients*	
8	2	26615		29							Intercept	26330.69093	
9	3	27039		29							TIME	508.003185	
10	4	40170		30									
11	1	23641		30							**Seasonal Averages**		
12	2	30477		31							(Actual/Forecast)		
13	3	30065		31							Quarter		
14	4	43805		32							1	0.777	
15	1	24256		32							2	0.947	
16	2	30976		33							3	0.923	
17	3	30579		33							4	1.353	
18	4	45471		34							Sum:	4.000	
19	1	25891		34									
20	2	33050		35							**Mean Sqauare Error:**		
21	3	33385		35982.8	0.928	33206.96	6746312.853	30982.64923			Linear Only		8505.569
22	4	50311		36490.8	1.379	49381.32	190999182.2	864306.0094			Seasonal Correction		2003.672
23	1	29177		36998.8	0.789	28753.89	61179895.26	179021.4011					

Solver Parameters

Set Target Cell: $E$1 — Solve

Equal To: Max / Min / Value of: 0 — Close

By Changing Cells: $M$8:$M$9,$M$14:$M$17 — Guess

Subject to the Constraints: $M$18 = 4 — Add, Change, Delete, Options, Reset All, Help

6.2.3. Make Up Your Mind: Live Decision Analysis. This fourth example comes from trying to overcome the fact that certain spreadsheet methods—such as sorting, pivot tables, and some statistical add-ins (regression)—are not "live," in the sense that if one changes the data, one has to take additional steps to resort and/or recalculate the results. One of the author's experiences as a decision analysis consultant led to the observation that in meetings, the focus tended to shift to the "artifacts" of decision analysis (e.g., redrawing trees, redoing distributions), which was often disruptive to the rhythm of the free flow of ideas and the meeting process. This spreadsheet was developed to overcome this limitation, and, thus, developed "live" methods for producing tornado charts (which involved developing a "live sort" method), probability distribution sculpting, and decision tree creation and analysis. This example involves two files: da1.xls and da2.xls.

This fourth example is based on the tool

(10) data tables.

Method 10: Data tables. In our opinion, data tables are another useful but "nonintuitive" Excel feature. Data tables are a "what-if" tool that allows users to explore the effect of systematically changing values in a formula. Data tables are as close to a "do loop" as Excel gets (without using VBA). Data tables are best explained with an example (see table.xls and Figure 8).

The spreadsheet calculates a loan payment amount for input variables such as interest rate, loan amount, etc. In the example, cells B3:B6 have input values, and cell B8 has the calculated loan amount. Cells A11:B20 are set up to be filled in as a one-variable data table. The column A12:A20 has possible interest rates. Cell B11 is a formula referring to the calculated payment amount. By selecting A11:B20 then going to *Data-Table*, a menu appears. Because it is the interest rate that we want to vary, and that is in a column, enter B3 as the column input cell, then click "OK." The payment amount for each variable then fills the table. Note that the formula for a one-variable table goes at the top of the column to be filled in.

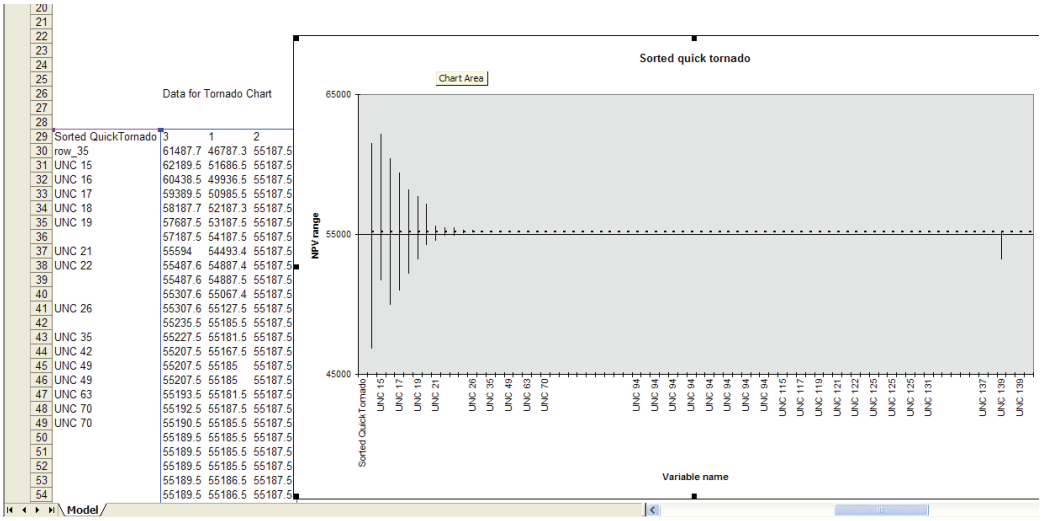
Cells D11:K20 are set up to be filled in as a two-way data table. The column D12:D20, again, has possible interest rate values. The row E11:K11 has possible payment amounts. Here, the reference for the output cell goes in the "corner" of the table (cell D11 here). Select D11:K20, enter B3 as the column input cell and B6 as the row input cell, and click OK. A payment table is produced.

The "live decision analysis" spreadsheets involve many of the "top 10" methods in the paper. This model takes as given a fairly typical financial model to be used as the basis for deterministic evaluation of scenarios. It then uses data tables (which respond "live" to changes in their underlying formulas) and graphical displays to examine the effect of changes on the input data.

FIGURE 8. Data table example.

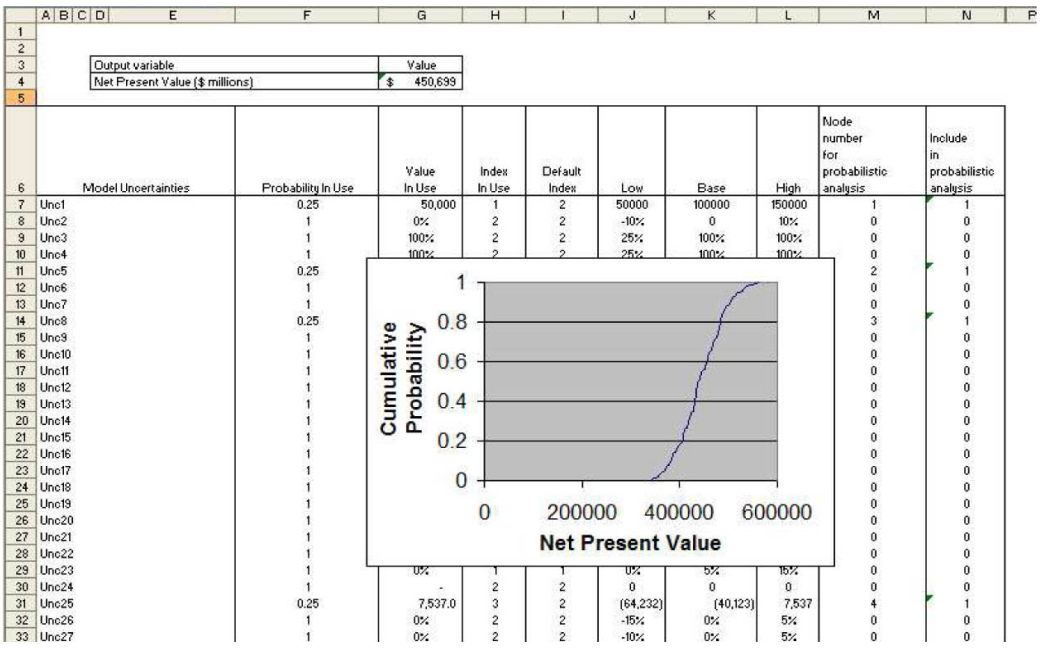
	A	B	C	D	E	F	G	H	I	J	K
1	Loan Payment Calculator										
2											
3	Annual Interest Rate:	8.00%									
4	# Payments per Year:	12									
5	Term of Loan (Years):	2									
6	Loan Amount:	$5,000									
7											
8	Payment:	$226.14									
9											
10	**One-Way Data Table**			**Two-Way Data Table**							
11	Interest Rate Effect:	$ 226.14		$226.14	$ 3,000	$ 3,500	$ 4,000	$ 4,500	$ 5,000	$ 5,500	$ 6,000
12	6.00%	$ 221.60		6.00%	$ 132.96	$ 155.12	$ 177.28	$ 199.44	$ 221.60	$ 243.76	$ 265.92
13	6.50%	$ 222.73		6.50%	$ 133.64	$ 155.91	$ 178.19	$ 200.46	$ 222.73	$ 245.00	$ 267.28
14	7.00%	$ 223.86		7.00%	$ 134.32	$ 156.70	$ 179.09	$ 201.48	$ 223.86	$ 246.25	$ 268.64
15	7.50%	$ 225.00		7.50%	$ 135.00	$ 157.50	$ 180.00	$ 202.50	$ 225.00	$ 247.50	$ 270.00
16	8.00%	$ 226.14		8.00%	$ 135.68	$ 158.30	$ 180.91	$ 203.52	$ 226.14	$ 248.75	$ 271.36
17	8.50%	$ 227.28		8.50%	$ 136.37	$ 159.09	$ 181.82	$ 204.55	$ 227.28	$ 250.01	$ 272.73
18	9.00%	$ 228.42		9.00%	$ 137.05	$ 159.90	$ 182.74	$ 205.58	$ 228.42	$ 251.27	$ 274.11
19	9.50%	$ 229.57		9.50%	$ 137.74	$ 160.70	$ 183.66	$ 206.62	$ 229.57	$ 252.53	$ 275.49
20	10.00%	$ 230.72		10.00%	$ 138.43	$ 161.51	$ 184.58	$ 207.65	$ 230.72	$ 253.80	$ 276.87
21											

FIGURE 9. Live decision analysis tornado chart.



Tornado charts display the effects of changes in input values from the largest to the smallest impact (see Figure 9), so "live" tornado charts require a "live" sorting procedure as well. The "live" sort relies heavily on rank and index functions (which are in the same family as the lookup functions previously discussed). The "live" probability distributions (see Figure 10) use mostly the same functions, and from them, we can also calculate value of information in real time. The "live" decision tree requires pivot tables as well. Once values for the endpoints of a decision tree are calculated, they are entered (not live) into a pivot table along with information about the sequence of events leading to each endpoint. Then, the process of "flipping the tree"—applying Bayes' rule to calculate conditional probability distributions under states of information—requires only the intuitive step of dragging columns so that they are in the same order as the event nodes in the version of the decision tree to be evaluated.

FIGURE 10. Live decision analysis probability distribution.



Live decision analysis can change the focus from deterministic models—for which analysis is used to derive other values—to those derived values themselves (e.g., value of information, option value, risk premium). By adjusting assumptions and decisions, it is then possible to actively sculpt a probability distribution. For example, a company might seek to maximize the value of information in a situation in which it expects to have exclusive access to that information, or it might seek to maximize the risk premium in a situation in which it has a higher risk tolerance than its competitors. This concept has facilitated rapid modeling for meta-decision making, such as decision process design and risk allocation. The application described here is meant to support such efforts. It has been used in classroom settings, where students have found it to have intuitive appeal. As an aside, we undertook this and other efforts in part to apply spreadsheet techniques in our own field as a challenge in itself to learn more about the capabilities of Excel—in this case, to find a use for such capabilities as pivot tables and sort functions. Because Excel is a platform for application development, rather than merely an application itself, this kind of experimenting is an effective (and fun) way to develop skills.

7. Learn More! Join Us! Help Us "Spread" the Good Word!

In this tutorial, we have explored both "sin" and "salvation" in the spreadsheet kingdom. We have discussed ways to enhance the effectiveness and efficiency of the spreadsheet development process, including principles of spreadsheet engineering and robust spreadsheet design. We have discussed a number of good spreadsheet practices and the Excel features that support these practices. Highlighted among these practices in the examples are

- the use of the methodology of: plan, build, test;
- separation of data, from analysis, from presentation; and
- the creative mixing of multiple analysis methods and innovative presentation methods.

The core of this tutorial, however, goes well beyond "tips and tricks"—the goal is to enable OR/MS professionals to harness the power of spreadsheets to support their particular areas of interest. Exploring spreadsheet functions and methods can spark new ideas for ways to implement OR/MS methodology and systems, while, in turn, new OR/MS methods spark the need for more "killer ap" spreadsheets.

Spreadsheets are certainly not the only tool for OR/MS model development, and we would never advocate that all work be done in spreadsheets. However, the advantages of spreadsheets, such as the ability to easily mix words, formulas, data, and graphs, as well as their flexibility, make them particularly appropriate for brainstorming and prototyping projects. One of the messages of this tutorial is, if spreadsheets are designed with purpose and care and if OR/MS developers take advantage of some of the advanced built-in (or added-in) capabilities, spreadsheets can be used for production applications as well.

If we have been successful with this tutorial, we have whetted your appetite for more. We encourage you to join SPRIG and become actively involved. Attend our sessions and conferences, share your own "killer aps," or even start your own spreadsheet research!

Acknowledgments

The authors thank Michael Johnson whose editorial wisdom and keen eye have greatly improved this chapter, the University at Albany and President Kermit Hall for their support of this endeavor, and SPRIG and Tom Grossman for focusing the attention of the OR/MS community on spreadsheets.

References

[1] S. C. Albright and W. L. Winston. *Spreadsheet Modeling and Applications: Essentials of Practical Management Science.* Southwestern College Publishing, Cincinnati, OH, 2004.

[2] A. A. Arens, R. J. Elder, and M. Beasley. *Auditing and Assurance Services: An Integrated Approach*, 11th ed. Prentice-Hall, Englewood Cliffs, NJ, 2005.

[3] B. Boehm and V. R. Basili. Software defect reduction top 10 list. *IEEE Computer* 34(1):135–137, 2001.

[4] R. Butler and D. Chadwick. Stop that subversive spreadsheet! EuSPRIG. http://www.eusprig.org/eusprig.pdf. 2003.

[5] R. J. Casimer. Real programmers don't use spreadsheets. *ACM SIGPLAN Notices* 27(6):10–16, 1993.

[6] R. B. Chase, F. R. Jacobs, and N. J. Aquilano. *Operations Management for Competitive Advantage*, 10th ed. McGraw Hill/Irwin, New York, 2004.

[7] G. Croll. The importance and criticality of spreadsheets in the City of London. D. Ward, ed. *EuSPRIG 2005 Conference Proceedings* 82–94, 2005.

[8] Data Analysis Story Library (DASL). http://lib.stat.cmu.edu/DASL/Stories/dealersales.html.

[9] EBI home page. http://www.webebi.com/.

[10] EuSPRIG home page. http://eusprig.org.

[11] T. A. Grossman, V. Mehrotra, and Özgür Özlük. Lessons from mission critical spreadsheets. Working paper, San Francisco School of Business and Management, San Francisco, CA, 2006.

[12] G. Harvey. *Excel 2003 for Dummies*. Wiley Publishing, Hoboken, NJ, 2003.

[13] S. McConnell. *Rapid Development*. Microsoft Press, Redmond, WA, 1996.

[14] S. McConnell. *Code Complete*, 2nd ed. Microsoft Press, Redmond, WA, 2004.

[15] Microsoft. Press release. http://www.microsoft.com/presspass/press/2003/oct03/10-13vstoofficelaunchpr.mspx. October 13, 2003.

[16] Microsoft. Statistical errors page. http://support.microsoft.com/default.aspx?kbid=828888&product=xl2003.

[17] P. O'Beirne. *Spreadsheet Check and Control*. Systems Publishing, Wexford, Ireland, 2005.

[18] S. G. Powell and K. R. Baker. *The Art of Modeling with Spreadsheets*. John Wiley & Sons, Danvers, MA, 2004.

[19] R. S. Pressman. *Software Engineering: A Practitioner's Approach*, 6th ed. McGraw-Hill, New York, 2005.

[20] C. Ragsdale. *Spreadsheet Modeling & Decision Analysis*, 5th ed. Southwestern College Publishing, Cincinnati, OH, 2006.

[21] L. R. Rittenberg and B. J. Schwieger. *Auditing: Concepts for a Changing Enviroment*, 5th ed. South-Western College Publishing, Cincinnati, OH, 2004.

[22] C. Scaffidi, M. Shaw, and B. Myers. Estimating the numbers of end users and end user programmers. *IEEE Symposium on Visual Languages and Human-Centric Computing* 207–214, 2005.

[23] M. H. Serif, R. K. Ahuja, and W. L. Winston. *Developing Spreadsheet-Based Decision Support Systems Using VBA for Excel*. Duxbury Press, Pacific Grove, CA, 2006.

[24] I. Sommerville. *Software Engineering*, 7th ed. Addision-Wesley, Boston, MA, 2004.

[25] SPRIG. http://sprig.section.informs.org/.

[26] Spreadsheet Research (SSR). http://panko.cba.hawaii.edu/ssr/.

[27] J. Tennent and G. Friend. *Guide to Business Modelling*. Bloomberg Press, London, UK, 2005.

[28] J. Walkenbach. *Excel 2003 Bible*. Wiley Publishing, Indianapolis, IN, 2003.

TUTORIALS IN
OPERATIONS RESEARCH
INFORMS 2006

 | ISBN 1-877640-20-9
DOI 10.1287/educ.1063.0026

Multiechelon Production/Inventory Systems: Optimal Policies, Heuristics, and Algorithms

Geert-Jan van Houtum
Department of Technology Management, Technische Universiteit Eindhoven, P.O. Box 513, 5600 MB, Eindhoven, The Netherlands, g.j.v.houtum@tm.tue.nl

Abstract The theory on multiechelon production/inventory systems is a core theory within supply chain management. It provides useful insights for design of supply chains and may be used for tactical and operational planning decisions. The multiechelon theory started with the seminal paper of Clark and Scarf in 1960. In this tutorial, we describe for which systems optimal policies are known, which key features are needed for these optimal policy structures, and we discuss heuristics for systems of which the optimal policy structure is not known. We describe the complete analysis for the most basic multiechelon production/inventory system: The serial, two-echelon production/inventory system with linear inventory holding and backordering costs. We show that base-stock policies are optimal, derive a decomposition result for the determination of optimal base-stock levels, present newsboy equations for the optimal base-stock levels, and discuss computational procedures. Next, we describe a variety of systems for which generalized classes of base-stock policies have been derived to be optimal. This includes assembly systems and systems with fixed batch sizes, fixed replenishment intervals, generalized demand processes, and a service-level constraint instead of backordering costs. Finally, we discuss approaches that have been taken for distribution systems and systems with a general structure.

Keywords production/inventory; multiechelon; stochastic demand; stochastic dynamic programming; base-stock policies; newsboy equations

1. Introduction

Supply chain management is a broad area that covers strategic, tactical, and operational management decisions. The objective of a supply chain is to deliver products of the right quality, at the right time, in the right amount, and, preferably, with low costs. Two primary sources of costs in supply chains are *capacity costs* and *material costs*. Typically, capacity decisions are made for a longer term than material decisions; thus, capacity decisions are often made first, and material decisions follow. Material decisions may also be made sequentially, according to a *hierarchical approach* with two decision levels.

(i) A *first level* decides on such things as the form of batching, the batch sizes and replenishment intervals, and the (planned) lead times, where a *multi-item, multiechelon view* is taken. Via these decisions, one can accommodate setups, capacity constraints, capacity partitioning, and shipment consolidation. These decisions may be reviewed annually, for example;

(ii) A *second level* decides on reorder and base-stock levels, adapted on a daily, weekly, or monthly basis (e.g., when procedures like exponential smoothing are used for demand forecasting). Here, the batching rule is taken as given, and a *single-item, multiechelon view* can be incorporated.

The essential feature of this approach is that batching decisions are separated from safety stock decisions, as advocated by Graves [39]. For the second-level material decisions, excellent support may be provided by *multiechelon production/inventory models*. In addition,

the multiechelon models give insights into the effect of lead times, batch sizes, and demand uncertainty on total costs. They, thus, may also support first-level material decisions, capacity decisions, and design decisions (see also de Kok and Graves [17], Tayur et al. [59]).

The theory of multiechelon production/inventory decisions is the topic of this chapter. This theory was started by Clark and Scarf [14] in 1960. In their paper, a basic model for a supply chain consisting of multiple stages with a serial structure is considered. The stages are numbered $1, \ldots, N$. Stage N orders at an external supplier, stage $N-1$ orders at stage N, stage $N-2$ orders at stage $N-1$, and so on. Finally, at the most downstream stage, stage 1, external demand occurs. A stage may represent a production node, in which case input material is transformed into another product, or a transportation node, in which case a product is moved from one location to another. At the end of each stage, products can be kept on stock in a stockpoint, where they stay until they are demanded by either the next stage or the external customers. Time consists of periods of equal length, which may be days, weeks, or months, and the time horizon is infinite. Each stage is allowed to order at the beginning of each period. One can never order more than the amount available at the supplying stage, and the ordered amount by a stage n is assumed to arrive at the stockpoint at the end of stage n after a deterministic lead time. For the demand, a stationary, stochastic demand process is assumed. Costs consist of (linear) inventory-holding costs, which models the costs of working capital in the supply chain, and linear penalty costs for backordering, which constitute the counterpart for the inventory-holding costs. Clark and Scarf proved that so-called base-stock policies based on echelon inventory positions are optimal, and they showed that the optimal base-stock levels are obtained by the minimization of one-dimensional convex cost functions (this is known as the decomposition result). We refer to their model as the Clark-Scarf model.

Since 1960, much research has been executed to extend the work of Clark and Scarf. Extensions that have been considered are systems with a pure assembly/convergent structure, fixed batch sizes or fixed replenishment intervals, a service-level constraint, and advance demand information. Also, alternative approaches were developed to derive the main results for the Clark-Scarf model, which has contributed to a better understanding of which features are key to obtain the optimality of base-stock policies.

The objective of this tutorial is to expose for which systems optimal policies are known, which key features are needed to be able to derive the structure of optimal policies, and to discuss heuristics for systems of which the optimal policy structure is not known. We will start with a complete analysis of the most basic system: The two-echelon, serial system. From there on, we describe many extensions that have been made. For these extensions, generalized forms of base-stock policies have been shown to be optimal. This includes assembly/convergent systems. For distribution/divergent systems, base-stock policies are optimal under the so-called balance assumption, but they are not optimal without that assumption.

Systems with a general structure (i.e., with a mixed convergent-divergent structure) are most difficult. For those systems, concepts have been developed based on base-stock policies, and those concepts can be related to insights for basic systems (see §5.3). In the past few years, these concepts have been successfully applied in practice. In de Kok et al. [18], Graves and Willems [40], and Lin et al. [46], applications in large-scale projects at IBM, Eastman Kodak, and Philips Electronics have been reported. There are also several applications in smaller projects, and, currently, there is also commercial software available that is based on multiechelon theory. Generally, multiechelon theory is increasingly incorporated into the practice of supply chain management.

The foreknowledge that we assume is basic probability theory, basic inventory theory (e.g., Axsäter [3], Zipkin [71]), and stochastic dynamic programming (e.g., Porteus [49], Puterman [50]). This tutorial is intended to be accessible for anyone with that foreknowledge. It may also serve as a starting point for a Ph.D. course on multiechelon production/inventory systems, and for starting researchers in this research area.

The organization is as follows. In §2, we give a complete treatment of a two-echelon, serial system, and we denote the key features that lead to the optimality of base-stock policies, the decomposition result, and newsboy equations for optimal base-stock levels. Next, in §3, we describe the generalized results for multiechelon, serial systems, and we discuss exact and approximate procedures for the computation of an optimal policy and the corresponding optimal costs. In §4, we describe a variety of model variants and extended models for which pure or generalized forms of base-stock policies are optimal. This includes assembly/convergent systems and systems with a service-level constraint, fixed batch sizes, and fixed replenishment intervals. Then, in §5, we discuss systems with a distribution/divergent structure and systems with a mixed convergent-divergent structure. After that, in §6, we classify multiechelon systems as nice and complicated systems, and we conclude.

2. Analysis of the Two-Echelon, Serial System

In this section, we give a complete analysis of the two-echelon, serial system. In §2.1, we describe the model. Next, in §2.2, we derive the optimality of base-stock policies under general convex echelon cost functions, and we show that the optimal base-stock levels follow from the minimization of convex, one-dimensional functions (this is known as the decomposition result). Subsequently, in §2.3, for the common case with linear inventory holding and penalty costs, we derive simpler expressions in terms of so-called shortfalls and backlogs for these convex, one-dimensional functions. These alternative expressions facilitate computational procedures, and we use them to derive newsboy equations for the optimal base-stock levels.

2.1. Model

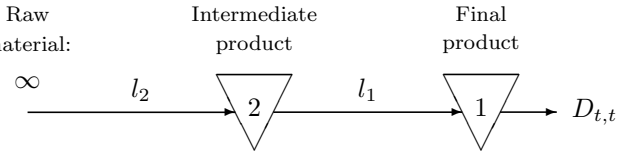
Consider a supply chain consisting of two stages, in which a single product is produced to stock. The upstream stage is called stage 2 and the downstream stage is called stage 1. Both stage 1 and stage 2 consists of a production step, a transportation step, or a network of such steps, with a stockpoint at the end of the stage. The stockpoint at the end of stage $n = 1, 2$ is called stockpoint n. For simplicity, we say that stage 2 is fed with raw materials, that an intermediate product is obtained from stage 2 and stored in stockpoint 2, and that a final product is obtained from stage 1 and stored in stockpoint 1. External demand occurs for the final product, i.e., at stockpoint 1.

Time is divided into periods of equal length. W.l.o.g., the length of each period, is assumed to be equal to 1. The time horizon that we consider is infinitely long. The periods are numbered $0, 1, \ldots$, and denoted by the index t ($t \in \mathbb{N}_0 := \{0\} \cup \mathbb{N}$).

Both stages or stockpoints are allowed to place orders at the beginning of each period. An amount ordered by stage 2 at the beginning of a period t arrives at stockpoint 2 after a deterministic lead time $l_2 \in \mathbb{N}$. We assume that sufficient raw material is always available, and, thus, orders by stockpoint 2 are never delayed. An amount ordered by stage 1 at the beginning of a period t arrives at stockpoint 1 after a deterministic lead time $l_1 \in \mathbb{N}_0$ ($l_1 = 0$ is allowed), provided that there is sufficient stock at stockpoint 2 available at the beginning of period t. If the available stock is smaller than the ordered amount, then the available amount is sent into stage 1 and becomes available after l_1 periods, while the rest is delivered as soon as possible.

The demands in different periods are independent and identically distributed on $[0, \infty)$. The cumulative demand over periods $t_1, \ldots, t_2$, $0 \le t_1 \le t_2$, is denoted by D_{t_1, t_2}. F is the generic distribution function for the demand $D_{t,t}$ in an arbitrary period $t \in \mathbb{N}_0$. The mean demand per period is $\mu > 0$. We implicitly assume that we have a continuous product and that order sizes and inventory levels are real-valued variables. The demand distribution function, however, is not necessarily continuous. There may be positive probability masses at specific points. In the case of a discrete product, it is more natural to limit order sizes and inventory levels to integer values. That case is discussed in §4.2.

FIGURE 1. The serial, two-echelon production/inventory system.



A picture of the serial, two-echelon system is given in Figure 1. We have the following events in each period.

(i) at each stage, an order is placed;
(ii) arrival of orders;
(iii) demand occurs; and
(iv) one-period costs are assessed (these costs are specified below).

The first two events take place at the beginning of the period, and the order of these two events may be interchanged, except for the most downstream stage when its lead time equals 0. The last event occurs at the end of a period. The third event, the demand, may occur anywhere in between.

2.1.1. Echelon Stocks and Costs Attached to Echelons. The analysis of multiechelon systems is generally based on the concepts *echelon stock* and *echelon inventory position*, as introduced by Clark [13] in 1958 (see also Zipkin [71], pp. 120–124). Below, we describe these concepts and define costs attached to echelons.

In general, the echelon stock (or *echelon inventory level*) of a given stockpoint denotes all physical stock at that stockpoint plus all materials in transit to or on hand at any stockpoint downstream minus eventual backlogs at the most downstream stockpoints. The chain under consideration is called the *echelon*. An echelon stock may be negative, indicating that the total backlog at the most downstream stockpoints is larger than the total physical stock in that echelon. Echelons are numbered according to the highest stockpoint in that echelon. In our two-echelon system, we have two echelons:

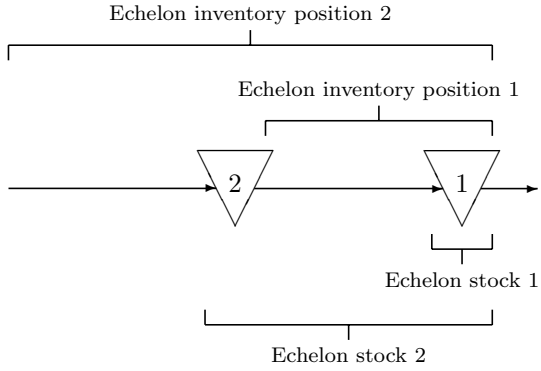
- echelon 1, consisting of stockpoint 1; and
- echelon 2, consisting of stockpoint 2, stockpoint 1, and the pipeline in between.

The echelon stock of echelon 1 is also called *echelon stock* 1, and is the same as the installation stock of stockpoint 1. The echelon stock of echelon 2 is also called *echelon stock* 2.

The *echelon inventory position* of a stockpoint is defined as its echelon stock plus all materials that are in transit to the stockpoint. We assume that a stockpoint never orders more than what is available at the next upstream stockpoint. In our two-echelon system, this implies that stockpoint 1 never orders more than what is available at stockpoint 2. As we study the optimal behavior of the system under centralized control, this assumption can be made w.l.o.g.; instead of creating a backlog position at stockpoint 2, stockpoint 1 will attempt to order that difference at the next period. Under this assumption, the echelon inventory position is also equal to the echelon stock plus all materials on order. The echelon inventory position of echelon n is also called *echelon inventory position* n, $n = 1, 2$. The echelon stocks and echelon inventory positions are visualized in Figure 2.

We now define our costs, which are assessed at the *end* of each period, based on the echelon stocks. For $n = 1, 2$, we pay costs $c_n(x_n)$, where x_n denotes echelon stock n at the *end* of a period. Notice that, by the above definitions, it holds that $x_2 \geq x_1$. The function $c_n(x_n)$ denotes the *costs attached to echelon* n. We assume that the cost functions $c_n(x_n)$, $n = 1, 2$, are *convex*. In addition, to avoid mathematical complexities, we assume that these cost functions are such that it is suboptimal to let the backlog grow to infinity. That one-period costs can be expressed as the sum of separable, convex functions based on echelon stocks is a crucial assumption. This was already pointed out by Clark and Scarf [14] (Assumption 3, pp. 478–479).

FIGURE 2. The concepts echelon stock and echelon inventory position.



A *special cost structure* is obtained when we have *linear inventory-holding and penalty costs*. That structure is often assumed and is as follows. A cost of $h_2 \geq 0$ is charged for each unit that is on stock in stockpoint 2 at the end of a period and for each unit in the pipeline from stockpoint 2 to stockpoint 1. A cost of $h_1 + h_2 \geq 0$ is charged for each unit that is on stock in stockpoint 1 at the end of a period. The inventory-holding cost parameters represent interest and storage costs. We assume that the additional inventory-holding cost at stage 1 is nonnegative, i.e., $h_1 \geq 0$. A penalty cost p is charged per unit of backordered demand at stockpoint 1 at the end of a period. This represents inconvenience for delayed fulfillment of demand and constitutes the counterpart for the inventory-holding costs. We assume that $p > 0$.

Let x_n, $n = 1, 2$, be echelon stock n at the end of a period. Then, the total inventory holding and backordering costs at the end of a period are equal to

$$h_2(x_2 - x_1) + (h_1 + h_2)x_1^+ + px_1^-,$$

where $x^+ = \max\{0, x\}$ and $x^- = \max\{0, -x\} = -\min\{0, x\}$ for any $x \in \mathbb{R}$. These costs may be rewritten as

$$\begin{aligned} &h_2(x_2 - x_1) + (h_1 + h_2)x_1^+ + px_1^- \\ &\quad = h_2(x_2 - x_1) + (h_1 + h_2)x_1 + (p + h_1 + h_2)x_1^- \\ &\quad = h_2 x_2 + h_1 x_1 + (p + h_1 + h_2)x_1^- \\ &\quad = c_2(x_2) + c_1(x_1), \end{aligned}$$

with

$$c_1(x_1) = h_1 x_1 + (p + h_1 + h_2)x_1^-, \tag{1}$$

$$c_2(x_2) = h_2 x_2. \tag{2}$$

This shows that the case with linear inventory holding and penalty costs fits under the general cost structure. In this special case, $c_2(x_2)$ is linear and $c_1(x_1)$ is a convex function consisting of two linear segments. In the analysis below (in §2.2), we assume the general cost structure. After that, we derive additional results that hold under linear inventory holding and penalty costs (in §2.3).

2.1.2. Objective. Let Π denote the set of all possible ordering policies, and let $G(\pi)$ denote the average costs of ordering policy π for all $\pi \in \Pi$. We want to solve the following minimization problem to optimality.

$$\begin{aligned} \text{(P):} \quad &\min G(\pi) \\ &\text{s.t.} \quad \pi \in \Pi. \end{aligned}$$

So, the objective is to find an ordering policy under which the average costs per period are minimized.

2.2. Analysis

In this subsection, we derive the optimality of base-stock policies and the decomposition result. These results are due to Clark and Scarf [14], who derived these results via a stochastic dynamic program in a finite-horizon setting. Federgruen and Zipkin [29] extended these results to the infinite-horizon case. Alternative, easier proofs were developed by Langenhoff and Zijm [45] and by Chen and Zheng [12] (see also Chen [10]). We follow the approach of Chen and Zheng, where we add an explicit definition of a relaxed single-cycle problem (cf. van Houtum et al. [66] for a generalized system; Chen and Zheng have an implicit definition). We distinguish three steps:

1. definition of cycles and cycle costs;
2. solution of a relaxed single-cycle problem; and
3. solution of the infinite-horizon problem (P).

These steps are described in §§2.2.1–2.2.3. The introduction of the relaxed single-cycle problem and the property that the solution of the single-cycle problem also solves the infinite-horizon problem (P) are key in the line of proof. Interestingly, the relaxed single-cycle problem is a stochastic dynamic programming problem with a finite number of stages (two stages in this case). Thus, the solution of problem (P), which is a stochastic dynamic programming problem with an infinite horizon, follows in fact from a finite-horizon stochastic programming problem.

2.2.1. Step 1: Definition of Cycles and Cycle Costs. We consider the connection between order decisions at the two stages, and we describe which costs they affect.

For each $n = 1, 2$ and $t \in \mathbb{N}_0$, let $IL_{t,n}$ and $IP_{t,n}$ denote echelon stock n (= echelon inventory level n) and echelon inventory position n at the beginning of period t (just before the demand occurs), and let $C_{t,n}$ denote the costs attached to echelon n at the end of period t.

We now consider the following two connected decisions, starting with an order placed by stage 2 at the beginning of a period $t_0 \in \mathbb{N}_0$:

- *Decision* 2: Decision 2 concerns the decision at the beginning of period t_0 with respect to the order placed by stage 2. Suppose that this order is such that $IP_{t_0,2}$ becomes equal to some level z_2. First of all, this decision directly affects the echelon 2 costs at the end of period $t_0 + l_2$. The expected value of these costs equals

$$\mathsf{E}\{C_{t_0+l_2,2} | IP_{t_0,2} = z_2\} = \mathsf{E}\{c_2(z_2 - D_{t_0,t_0+l_2})\}. \tag{3}$$

Second, by this decision, echelon stock 2 at the beginning of period $t_0 + l_2$ becomes equal to $IL_{t_0+l_2,2} = z_2 - D_{t_0,t_0+l_2-1}$, and this directly limits the level to which one can increase the echelon inventory position $IP_{t_0+l_2,1}$ of echelon 1 at the beginning of period $t_0 + l_2$. This is the second decision to consider.

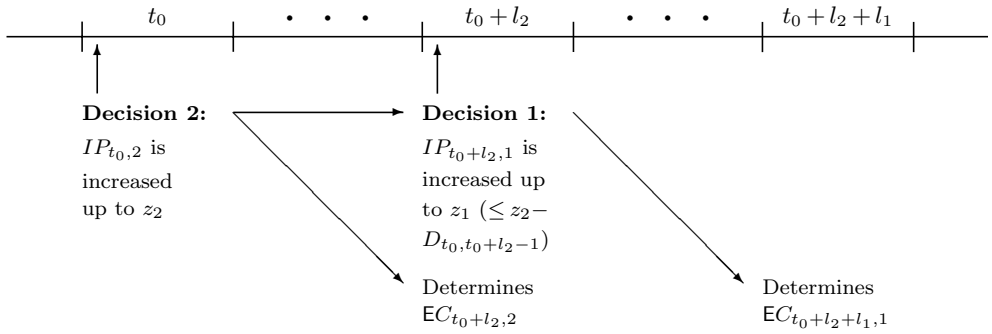
- *Decision* 1: Decision 1 concerns the order placed by stage 1 at the beginning of period $t_0 + l_2$. Suppose that by this order, $IP_{t_0+l_2,1}$ becomes equal to some level z_1. This decision directly affects the echelon 1 costs at the end of period $t_0 + l_2 + l_1$. The expected value of these costs equals

$$\mathsf{E}\{C_{t_0+l_2+l_1,1} | IP_{t_0+l_2,1} = z_1\} = \mathsf{E}\{c_1(z_1 - D_{t_0+l_2,t_0+l_2+l_1})\}. \tag{4}$$

Figure 3 visualizes the way in which the above decisions affect each other, and which costs are determined by them.

In the description above, we have explicitly described for decision 1 how the level z_1 to which $IP_{t_0+l_2,1}$ is increased is bounded from above. We will need this in the analysis below. Obviously, for both decisions 2 and 1, it also holds that the levels z_2 and z_1 to which $IP_{t_0,2}$ and $IP_{t_0+l_2,1}$ are increased, are bounded from below (by the level that one already has for

FIGURE 3. The consequences of the decisions 1 and 2.



its echelon inventory position just before the new order is placed). In the analysis below, this is taken into account too. But, this bounding from below will appear to be less important.

The decisions 2 and 1 start with decision 2 taken in period t_0. These decisions constitute a *cycle*, and the corresponding expected costs are equal to

$$C_{t_0} := C_{t_0+l_2,2} + C_{t_0+l_2+l_1,1}.$$

These costs are defined for each period $t_0 \in \mathbb{N}_0$, and we call them the *total costs attached to cycle* t_0. For each positive recurrent policy $\pi \in \Pi$, the average costs are equal to the average value of the costs C_{t_0} over all cycles t_0.

$$\begin{aligned} G(\pi) &= \lim_{T\to\infty} \frac{1}{T} \mathsf{E}\left\{ \sum_{t=0}^{T-1} (C_{t,2} + C_{t,1}) \right\} \\ &= \lim_{T\to\infty} \frac{1}{T} \mathsf{E}\left\{ \sum_{t=0}^{T-1} C_t + \sum_{t=0}^{l_2-1} C_{t,2} + \sum_{t=0}^{l_2+l_1-1} C_{t,1} - \sum_{t=T}^{T+l_2-1} C_{t,2} - \sum_{t=T}^{T+l_2+l_1-1} C_{t,1} \right\} \\ &= \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathsf{E}C_t \end{aligned} \tag{5}$$

2.2.2. Step 2: Solution of a Relaxed Single-Cycle Problem. Consider a cycle t_0. We now consider how the decisions 1 and 2 can be taken such that the expected total costs attached to cycle t_0 ($= \mathsf{E}C_{t_0}$) are minimized. Decision n, $n = 1, 2$, is described by the level z_n, to which echelon inventory position n is increased at the beginning of period t_0 and $t_0 + l_2$, respectively. The choice for the level z_1 is limited from above by what is available at stage 2. Further, the choice for the level z_n, $n = 2, 1$, is bounded from below by the value of echelon inventory position n just before the order is placed. By neglecting the bounding from below, we obtain the following *relaxed problem*:

$$\begin{aligned} (\mathrm{RP}(t_0)): \quad \text{Min} \quad & \mathsf{E}C_{t_0} = \mathsf{E}C_{t_0+l_2,2} + \mathsf{E}C_{t_0+l_2+l_1,1} \\ \text{s.t.} \quad & \mathsf{E}C_{t_0+l_2,2} = \mathsf{E}\{c_2(z_2 - D_{t_0,t_0+l_2})\}, \\ & \mathsf{E}C_{t_0+l_2+l_1,1} = \mathsf{E}\{c_1(z_1 - D_{t_0+l_2,t_0+l_2+l_1})\}, \\ & z_1 \leq IL_{t_0+l_2,2}, \\ & IL_{t_0+l_2,2} = z_2 - D_{t_0,t_0+l_2-1}. \end{aligned}$$

Problem $(\mathrm{RP}(t_0))$ is a two-stage stochastic dynamic programming problem. Decision 2 is described by z_2 and is not limited at all; we, thus, may connect this decision to a dummy starting state. The resulting direct expected costs are equal to $\mathsf{E}\{c_2(z_2 - D_{t_0,t_0+l_2})\}$. Decision 1 is described by z_1, and, via the constraint $z_1 \leq IL_{t_0+l_2,2}$, its decision space depends on the echelon stock 2 at the beginning of period $t_0 + l_2$, i.e., on $IL_{t_0+l_2,2}$. Hence, we use

$IL_{t_0+l_2,2}$ to describe the state of the system when decision 1 is taken. This state depends on decision 2 via the relation $IL_{t_0+l_2,2} = z_2 - D_{t_0,t_0+l_2-1}$. Decision 1 results in direct expected costs $\mathsf{E}\{c_1(z_1 - D_{t_0+l_2,t_0+l_2+l_1})\}$.

For problem (RP(t_0)), we first determine what is optimal for decision 1, and after that we consider decision 2.

Let the function $G_1(y_1)$ be defined by

$$G_1(y_1) := \mathsf{E}\{c_1(y_1 - D_{t_0+l_2,t_0+l_2+l_1})\}, \quad y_1 \in \mathbb{R}. \tag{6}$$

This function denotes the expected costs attached to echelon 1 at the end of a period $t_0 + l_1 + l_2$ if echelon inventory position 1 at the beginning of period $t_0 + l_2$ (i.e., l_1 periods earlier) has been increased up to level y_1.

Lemma 1 (On the Optimal Choice for z_1). *It holds that*
(i) *$G_1(y_1)$ is convex as a function of y_1, $y_1 \in \mathbb{R}$.*
(ii) *Let S_1 ($\in \mathbb{R} \cup \{\infty\}$) be chosen such that*

$$S_1 := \arg\min_{y_1 \in \mathbb{R}} G_1(y_1).$$

Then, for the problem ($RP(t_0)$), it is optimal to choose the level z_1 equal to S_1, or as high as possible if this level cannot be reached.

Proof. The formula for $G_1(y_1)$ may be rewritten as

$$G_1(y_1) = \int_0^\infty c_1(y_1 - x)\mathrm{d}F_{l_1+1}(x),$$

where F_{l_1+1} is the (l_1+1)-fold convolution of F. Let y_1^1, $y_1^2 \in \mathbb{R}$, and $\alpha \in [0,1]$, then, by the convexity of $c_1(\cdot)$,

$$\begin{aligned} G_1(\alpha y_1^1 + (1-\alpha)y_1^2) &= \int_0^\infty c_1(\alpha(y_1^1 - x) + (1-\alpha)(y_1^2 - x))\mathrm{d}F_{l_1+1}(x) \\ &\le \int_0^\infty [\alpha c_1(y_1^1 - x) + (1-\alpha)c_1(y_1^2 - x)]\mathrm{d}F_{l_1+1}(x) \\ &= \alpha G_1(y_1^1) + (1-\alpha)G_1(y_1^2), \end{aligned}$$

and, thus, $G_1(y_1)$ is convex. This proves Part (i).

Next, S_1 is defined as the point where $G_1(y_1)$ is minimized. If there are multiple points where $G_1(y_1)$ is minimized, then S_1 may be taken equal to any of these points. We can now show how decision 1, i.e., the choice for z_1, may be optimized for problem (RP(t_0)). This decision is taken at the beginning of period $t_0 + l_2$, and the choice for z_1 is bounded from above by $IL_{t_0+l_2,2}$. This decision only affects the costs $\mathsf{E}C_{t_0+l_2+l_1,1}$, which, by (6), are equal to $G_1(z_1)$. As the function G_1 is convex, these costs are minimized by choosing z_1 equal to $z_1 = S_1$ if $IL_{t_0+l_2,2} \ge S_1$, and equal to $z_1 = IL_{t_0+l_2,2}$ if $IL_{t_0+l_2,2} < S_1$. This completes the proof of Part (ii). □

By Lemma 1, for decision 1, it is optimal to apply *base-stock policy* S_1 (i.e., a base-stock policy with base-stock level S_1). Let $G_2(y_1, y_2)$ be defined as the expected cycle costs when a base-stock policy with level $y_2 \in \mathbb{R}$ is applied for decision 2 and a base-stock policy $y_1 \in \mathbb{R}$ for decision 1 (notice that we allow that $y_2 < y_1$ and y_1 and y_2 may also be negative). Then, $z_2 = y_2$, as the external supplier can always deliver, and for z_1, we find

$$z_1 = \min\{IL_{t_0+l_2,2}, y_1\} = \min\{y_2 - D_{t_0,t_0+l_2-1}, y_1\}.$$

Hence,

$$\begin{aligned} G_2(y_1,y_2) = \mathsf{E}\{&c_2(y_2 - D_{t_0,t_0+l_2}) + c_1(\min\{y_2 - D_{t_0,t_0+l_2-1}, y_1\} - D_{t_0+l_2,t_0+l_2+l_1})\}, \\ &y_1, y_2 \in \mathbb{R}. \end{aligned} \tag{7}$$

Lemma 2 (On the Optimal Choice for z_2). *It holds that*
(i) $G_2(S_1, y_2)$ *is convex as a function of* y_2, $y_2 \in \mathbb{R}$.
(ii) *Let* S_2 $(\in \mathbb{R} \cup \{\infty\})$ *be chosen such that*

$$S_2 := \arg\min_{y_2 \in \mathbb{R}} G_2(S_1, y_2).$$

Then, for problem $(RP(t_0))$, *it is optimal to choose the level* z_2 *equal to* S_2.

Proof. Let F_{l_2} be the l_2-fold convolution of F. The formula for $G_2(S_1, y_2)$ may be rewritten as

$$\begin{aligned} G_2(S_1, y_2) &= \mathsf{E}\{c_2(y_2 - D_{t_0, t_0+l_2})\} + \int_0^\infty \mathsf{E}\{c_1(\min\{y_2 - x, S_1\} - D_{t_0+l_2, t_0+l_2+l_1})\}\, \mathrm{d}F_{l_2}(x) \\ &= \mathsf{E}\{c_2(y_2 - D_{t_0, t_0+l_2})\} + G_1(S_1) + \int_0^\infty [G_1(\min\{y_2 - x, S_1\}) - G_1(S_1)]\, \mathrm{d}F_{l_2}(x) \\ &= \mathsf{E}\{c_2(y_2 - D_{t_0, t_0+l_2})\} + G_1(S_1) + \int_0^\infty \widetilde{G}_1(y_2 - x)\, \mathrm{d}F_{l_2}(x), \end{aligned} \tag{8}$$

where

$$\widetilde{G}_1(y) = G_1(\min\{y, S_1\}) - G_1(S_1) = \begin{cases} G_1(y) - G_1(S_1) & \text{if } y < S_1, \\ 0 & \text{if } y \geq S_1. \end{cases}$$

Because $G_1(\cdot)$ is convex, with a minimum in S_1, also the function $\widetilde{G}_1(y)$ is convex. Hence, along the same lines as for Part (i) of Lemma 1, the first and third term in (8) may be shown to be convex. This implies that $G_2(S_1, y_2)$ is convex as a function of y_2, which completes the proof of Part (i).

Next, S_2 is defined as the point that minimizes $G_2(S_1, y_2)$ as a function of y_2. If there are multiple points where $G_2(S_1, y_2)$ is minimized, then S_2 may be taken equal to any of these points. We can now show how decision 2, i.e., the choice for z_2, may be optimized for problem $(\mathrm{RP}(t_0))$. This decision is taken at the beginning of period t_0. This decision affects the costs $\mathsf{E}C_{t_0+l_2, 2}$ and $\mathsf{E}C_{t_0+l_2+l_1, 1}$. Whatever choice is made for z_2, it is optimal to take decision 1 according to a base-stock policy with base-stock level S_1 (by Part (ii) of Lemma 1). Hence, by (7),

$$\mathsf{E}C_{t_0+l_2, 2} + \mathsf{E}C_{t_0+l_2+l_1, 1} = G_2(S_1, y_2).$$

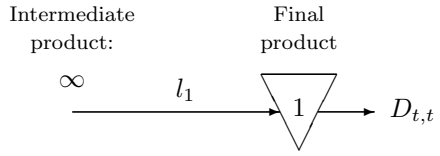
These costs are minimized by choosing z_2 equal to $z_2 = S_2$. This completes the proof of Part (ii). □

By Lemmas 1 and 2, for decisions 2 and 1 of problem $(\mathrm{RP}(t_0))$, it is optimal to apply a base-stock policy with base-stock level S_2 and S_1, respectively. The corresponding optimal costs are equal to $G_2(S_1, S_2)$. Because this problem was obtained by neglecting the bounding from below when placing orders, the optimal costs $G_2(S_1, S_2)$ constitute a lower bound for the optimal costs of the original problem (P).

2.2.3. Step 3: Solution of the Infinite-Horizon Problem (P). The functions $G_1(y_1)$ and $G_2(y_1, y_2)$ as defined above, have alternative interpretations; $G_1(y_1)$ represents the average costs of a base-stock policy y_1 in a specific single-echelon system, called *subsystem* 1, while $G_2(y_1, y_2)$ represents the average costs of a base-stock policy (y_1, y_2) for the full two-echelon system as depicted in Figure 1. This is shown first, and after that, we solve the original problem (P).

Subsystem 1 is defined as the downstream part of the full two-echelon system. It consists of stage 1 only, and it has stockpoint 2 as external supplier with infinite supply. The lead time for this subsystem is l_1, the demands are the same as in the full system, and the costs consist of the costs attached to echelon 1; see Figure 4. Under a base-stock policy y_1

FIGURE 4. Subsystem 1.



($y_1 \in \mathbb{R}$), at the beginning of each period, nothing is ordered if the current inventory position is already at level y_1 or higher, and the inventory position is increased up to level y_1 if the current inventory position is lower than y_1. That the inventory position before ordering is above the base-stock level y_1 may only happen in a limited number of periods. Hence, in steady state, the inventory position is always increased up to level y_1, and, therefore, the average costs are equal to $G_1(y_1) = \mathsf{E}\{c_1(y_1 - D_{t_0+l_2,\, t_0+l_2+l_1})\}$.

Consider now a base-stock policy (y_1, y_2), $y_1, y_2 \in \mathbb{R}$, for the full two-echelon system (we allow that $y_2 < y_1$). Under this policy, at the beginning of each period, stage 2 orders nothing if the current echelon inventory position 2 is already at level y_2 or higher, and its echelon inventory position is increased up to level y_2 if the current position is lower than y_2. That echelon inventory position 2 before ordering is above the base-stock level y_2 may only happen in a limited number of periods. Hence, in steady state, echelon inventory position 2 is always increased up to level y_2. Similarly, at the beginning of each period, stage 1 orders nothing if the current echelon inventory position 1 is already at level y_1 or higher, and, one aims to increase up to level y_1 if the current position is lower than y_1. In the latter case, it may not be possible to increase up to y_1 because there is not sufficient material available in stockpoint 2. That echelon inventory position 1 before ordering is above the base-stock level y_1 may only happen in a limited number of periods. Hence, in steady state, we obtain that echelon inventory position 1 is increased up to level y_1 if echelon stock 2 is at least y_1 at that moment, and up to echelon stock 2 otherwise. Hence, in steady state, we obtain per cycle that the ordering behavior is precisely as depicted in Figure 3 in which base-stock policies with levels y_2 and y_1 are applied for decisions 2 and 1, respectively. Hence, the average costs of a base-stock policy (y_1, y_2) are given by the function $G_2(y_1, y_2)$.

Suppose now that base-stock policy (S_1, S_2) is used for the original problem (P). Then average costs $G_2(S_1, S_2)$ are obtained, and these costs are, thus, equal to the lower bound. This implies that base-stock policy (S_1, S_2) is optimal for the original problem (P). In fact, if base-stock policy (S_1, S_2) is used in all periods, then the lower bounds that were relaxed in problem (RP(t_0)) are only binding during a transient period (when the echelon inventory positions may be above S_1 and S_2, and nothing should be ordered). In the long run, these lower bounds are not binding and, thus, the optimal solutions of the relaxed and unrelaxed problem are identical.

Theorem 1. *Base-stock policy (S_1, S_2), with the S_i as defined in Lemmas 1 and 2, is optimal for problem (P).*

This theorem shows that the class of base-stock policies is optimal, and that the optimal base-stock levels can be obtained sequentially by the minimization of one-dimensional functions. The latter result is known as the *decomposition result.*

Notice that it may happen that $S_2 < S_1$. As stated above, for base-stock policies (y_1, y_2) in general, we allow that $y_2 < y_1$, i.e., that the base-stock level for echelon inventory position 1 is larger than the base-stock level for echelon inventory position 2. Nevertheless, in practice, it is more natural to use and communicate a base-stock policy (y_1, y_2) with $y_2 \geq y_1$. The following lemma shows that any base-stock policy (y_1, y_2) with $y_2 < y_1$ can be translated into a base-stock policy $(\tilde{y}_1, y_2)$ with $y_2 \geq \tilde{y}_1$ and equal average costs.

Lemma 3. *Let $y_1, y_2 \in \mathbb{R}$, and define $\tilde{y}_1 := \min\{y_1, y_2\}$. Then $G_2(\tilde{y}_1, y_2) = G_2(y_1, y_2)$.*

Proof. Let $y_1, y_2 \in \mathbb{R}$, and define $\tilde{y}_1 := \min\{y_1, y_2\}$. That $G_2(\tilde{y}_1, y_2) = G_2(y_1, y_2)$ is trivial in case $y_2 \geq y_1$, because then $\tilde{y}_1 = y_1$. In case $y_2 < y_1$, at the beginning of each period, stage 1 is confronted with a shortage of material at stockpoint 2, and all available material at stockpoint 2 will be forwarded into stage 2. This implies that stockpoint 2 is a stockless stockpoint. This will still be so if base-stock level y_1 is decreased to $\tilde{y}_1 = y_2$. Hence, under base-stock policy $(\tilde{y}_1, y_2)$, the orders are identical to the orders generated under base-stock policy (y_1, y_2) (at least in the long run; in the first periods of the horizon, there may be differences). Thus, both policies have the same average costs. An alternative, technical proof is obtained by (7): If $y_2 < y_1$, then

$$\begin{aligned} G_2(y_1, y_2) &= \mathsf{E}\{c_2(y_2 - D_{t_0, t_0+l_2}) + c_1(y_2 - D_{t_0, t_0+l_2-1} - D_{t_0+l_2, t_0+l_2+l_1})\} \\ &= G_2(y_2, y_2) = G_2(\tilde{y}_1, y_2). \quad \square \end{aligned}$$

This completes the whole analysis for the two-echelon serial system. All results are easily extended to serial systems with more than two stages. Proofs go by induction, where the induction step is identical to what we derived for stage 2 in this two-echelon system.

Remark 1 (Induced Penalty Cost Function). Equation (8) for $G_2(S_1, y_2)$ consists of three terms. The first term denotes the costs attached to echelon 2. The second term, $G_1(S_1)$, denotes the minimal costs for subsystem 1. The third term denotes the additional costs when echelon stock 2 is insufficient to increase echelon inventory position 1 to its optimal value S_1. We defined S_2 as the point where $G_2(S_1, y_2)$ is minimized. Obviously, one finds the same optimal base-stock level by the minimization of the echelon 2 costs (the first term) plus the third term. This is how Clark and Scarf proceeded, and they interpreted the third term as an *induced penalty cost function.*

2.3. Linear Inventory Holding and Penalty Costs

In this subsection, *we assume that the echelon cost functions* $c_n(\cdot)$*,* $n = 1, 2$*, are given by* (1)–(2), i.e., we consider the special, but common, cost structure consisting of linear inventory holding and penalty costs. We derive interesting, additional results. First, in §2.3.1, we derive an alternative formula in terms of expected shortfalls and backlogs for the average costs of a base-stock policy. That formula facilitates computational procedures, and we exploit that formula to get the partial derivative to the base-stock level of echelon 2. For the average costs in subsystem 1, we obtain also a partial derivative, and the combination of both partial derivatives leads to newsboy equations for the optimal base-stock levels; see §2.3.2.

2.3.1. Alternative Cost Formulas for Base-Stock Policies. Assume the echelon costs functions as given by (1)–(2) and consider a base-stock policy (y_1, y_2), $y_1, y_2 \in \mathbb{R}$. The average costs $G_2(y_1, y_2)$ may be obtained by a single-cycle analysis; see Figure 3. The costs consist of the terms $C_{t_0+l_2, 2}$ and $C_{t_0+l_2+l_1, 1}$. The expected value of the costs $C_{t_0+l_2, 2}$ equals

$$\mathsf{E}C_{t_0+l_2, 2} = \mathsf{E}\{c_2(y_2 - D_{t_0, t_0+l_2})\} = \mathsf{E}\{h_2(y_2 - D_{t_0, t_0+l_2})\} = h_2(y_2 - (l_2 + 1)\mu).$$

Next, we study $\mathsf{E}C_{t_0+l_2+l_1, 1}$. The level z_1 denotes the actual level to which $IP_{t_0+l_2, 1}$ is increased. The difference with the desired level y_1 is called the *shortfall*, which can also be seen as a "backlog" at stockpoint 2 (it would be the backlog at stockpoint 2 if stage 1 would order such that $IP_{t_0+l_2, 1}$ is increased up to y_1, without taking into account how much is available at stockpoint 2). We denote this shortfall by B_1. This shortfall is equal to

$$\begin{aligned} B_1 &= y_1 - z_1 \\ &= y_1 - \min\{y_2 - D_{t_0, t_0+l_2-1}, y_1\} \\ &= y_1 + \max\{-y_2 + D_{t_0, t_0+l_2-1}, -y_1\} \\ &= \max\{0, y_1 - y_2 + D_{t_0, t_0+l_2-1}\} \\ &= (D_{t_0, t_0+l_2-1} - (y_2 - y_1))^+ \end{aligned} \tag{9}$$

(notice that by definition this shortfall is positive if $y_1 > y_2$). Now, define B_0 as the backlog at stockpoint 1 at the end of period $t_0 + l_2 + l_1$. Given that $IP_{t_0+l_2,1}$ is increased up to $z_1 = y_1 - B_1$, B_0 becomes equal to

$$\begin{aligned} B_0 &= (z_1 - D_{t_0+l_2,\,t_0+l_2+l_1})^- \\ &= (D_{t_0+l_2,\,t_0+l_2+l_1} - z_1)^+ \\ &= (D_{t_0+l_2,\,t_0+l_2+l_1} - (y_1 - B_1))^+ \\ &= (B_1 + D_{t_0+l_2,\,t_0+l_2+l_1} - y_1)^+. \end{aligned} \tag{10}$$

Then, for the costs attached to echelon 1 at the end of period $t_0 + l_2 + l_1$, we obtain

$$\begin{aligned} \mathsf{E}C_{t_0+l_2+l_1,1} &= \mathsf{E}\{c_1(z_1 - D_{t_0+l_2,\,t_0+l_2+l_1})\} \\ &= \mathsf{E}\{h_1(z_1 - D_{t_0+l_2,\,t_0+l_2+l_1}) + (p+h_1+h_2)(z_1 - D_{t_0+l_2,\,t_0+l_2+l_1})^-\} \\ &= h_1(y_1 - \mathsf{E}B_1 - (l_1+1)\mu) + (p+h_1+h_2)\mathsf{E}B_0. \end{aligned}$$

As a result, we find the following theorem. (The formula in this theorem stems from van Houtum and Zijm [62], where an equivalent formula has been derived, but with $\mathsf{E}B_1$ and $\mathsf{E}B_0$ expressed in integral form.)

Theorem 2. *Let the echelon cost functions $c_n(\cdot)$ be given by (1)–(2). Then, the average costs of a base-stock policy (y_1, y_2), with $y_1, y_2 \in \mathbb{R}$, are equal to*

$$G_2(y_1, y_2) = h_2(y_2 - (l_2+1)\mu) + h_1(y_1 - \mathsf{E}B_1 - (l_1+1)\mu) + (p+h_1+h_2)\mathsf{E}B_0,$$

where the random variables B_1 and B_0 are given by (9)–(10).

The formula for the average costs of a base-stock policy (y_1, y_2) also shows what the average backlog and average stock levels are. The term $\mathsf{E}B_0$ denotes the average backlog at the end of a period. The amount $y_1 - \mathsf{E}B_1 - (l_1+1)\mu + \mathsf{E}B_0$ is the average physical stock of echelon 1 (= stockpoint 1) at the end of a period; this is the amount for which a cost h_1 is paid per unit of product. The amount $y_2 - (l_2+1)\mu + \mathsf{E}B_0$ is the average physical stock of echelon 2 at the end of a period; this is the amount for which a cost h_2 is paid per unit of product. Further, the average stock in the pipeline between stockpoint 2 and stockpoint 1 is $l_1\mu$ (the throughput of the pipeline is equal to the mean demand and each unit of product is l_1 periods in the pipeline). This implies that the average physical stock in stockpoint 2 at the end of a period is equal to

$$\begin{aligned} &[y_2 - (l_2+1)\mu + \mathsf{E}B_0] - [y_1 - \mathsf{E}B_1 - (l_1+1)\mu + \mathsf{E}B_0] - l_1\mu \\ &\quad = y_2 - y_1 - l_2\mu + \mathsf{E}B_1 = \mathsf{E}\{((y_2 - y_1) - D_{t_0,\,t_0+l_2-1})^+\}. \end{aligned} \tag{11}$$

For the average costs in subsystem 1, under a base-stock policy y_1, $y_1 \in \mathbb{R}$, we find the following alternative expression (via (6)):

$$G_1(y_1) = h_1(y_1 - (l_1+1)\mu) + (p+h_1+h_2)\mathsf{E}B_0^{(1)}, \tag{12}$$

where the random variable $B_0^{(1)}$ represents the backlog in subsystem 1:

$$B_0^{(1)} = (D_{t_0+l_2,\,t_0+l_2+l_1} - y_1)^+. \tag{13}$$

Formula (12) shows that $G_1(y_1)$ is a newsboy function. Notice that $B_0^{(1)}$ is related to B_1 and B_0 in the following way: $B_0^{(1)} = (B_0 | B_1 = 0)$.

2.3.2. Newsboy Equations. We now determine the partial derivatives of $G_1(y_1)$ and $G_2(y_1, y_2)$. The derivative of $G_1(y_1)$ is denoted by $g_1(y_1)$. By (12),

$$g_1(y_1) = h_1 + (p + h_1 + h_2)\frac{\delta}{\delta y_1}\{\mathsf{E}B_0^{(1)}\}.$$

It is easily seen that

$$\frac{\delta}{\delta y_1}\{\mathsf{E}B_0^{(1)}\} = -\mathsf{P}\{B_0^{(1)} > 0\}.$$

Substitution of this property into the previous equation shows that

$$g_1(y_1) = h_1 - (p + h_1 + h_2)\mathsf{P}\{B_0^{(1)} > 0\}, \tag{14}$$

where $B_0^{(1)}$ is given by (13).

For the function $G_2(y_1, y_2)$, we are interested in the partial derivative with respect to the last component y_2. Hence, we define

$$g_2(y_1, y_2) := \frac{\delta}{\delta y_2}\{G_2(y_1, y_2)\}, \quad y_1, y_2 \in \mathbb{R}.$$

We find that

$$\begin{aligned} g_2(y_1, y_2) &= h_2 - h_1\frac{\delta}{\delta y_2}\{\mathsf{E}B_1\} + (p + h_1 + h_2)\frac{\delta}{\delta y_2}\{\mathsf{E}B_0\} \\ &= h_2 + h_1\mathsf{P}\{B_1 > 0\} - (p + h_1 + h_2)\mathsf{P}\{B_1 > 0 \text{ and } B_0 > 0\}. \end{aligned} \tag{15}$$

Here, the second step follows from the following properties.

$$\begin{aligned} \frac{\delta}{\delta y_2}\{\mathsf{E}B_1\} &= -\mathsf{P}\{B_1 > 0\}, \\ \frac{\delta}{\delta y_2}\{\mathsf{E}B_0\} &= -\mathsf{P}\{B_1 > 0 \text{ and } B_0 > 0\}. \end{aligned}$$

These properties are easily verified. The result in (15) constitutes the basis for the following lemma.

Lemma 4. *Let the echelon cost functions $c_n(\cdot)$ be given by* (1)–(2). *Then*

$$g_2(y_1, y_2) = (h_1 + h_2) - (p + h_1 + h_2)\mathsf{P}\{B_0 > 0\} - \mathsf{P}\{B_1 = 0\}g_1(y_1), \quad y_1, y_2 \in \mathbb{R},$$

with B_1 and B_0 given by (9)–(10).

Proof. It holds that

$$\begin{aligned} \mathsf{P}\{B_1 > 0\} &= 1 - \mathsf{P}\{B_1 = 0\}, \\ \mathsf{P}\{B_1 > 0 \text{ and } B_0 > 0\} &= \mathsf{P}\{B_0 > 0\} - \mathsf{P}\{B_1 = 0 \text{ and } B_0 > 0\} \\ &= \mathsf{P}\{B_0 > 0\} - \mathsf{P}\{B_0 > 0 | B_1 = 0\}\mathsf{P}\{B_1 = 0\}. \end{aligned}$$

By substitution of these expressions into Equation (15), we obtain (use the property that $B_0^{(1)} = (B_0 | B_1 = 0)$, and (14)):

$$\begin{aligned} g_2(y_1, y_2) &= h_2 + h_1(1 - \mathsf{P}\{B_1 = 0\}) \\ &\quad - (p + h_1 + h_2)(\mathsf{P}\{B_0 > 0\} - \mathsf{P}\{B_0 > 0 | B_1 = 0\}\mathsf{P}\{B_1 = 0\}) \\ &= (h_1 + h_2) - (p + h_1 + h_2)\mathsf{P}\{B_0 > 0\} \\ &\quad - \mathsf{P}\{B_1 = 0\}\left[h_1 - (p + h_1 + h_2)\mathsf{P}\{B_0 > 0 | B_1 = 0\}\right] \\ &= (h_1 + h_2) - (p + h_1 + h_2)\mathsf{P}\{B_0 > 0\} \\ &\quad - \mathsf{P}\{B_1 = 0\}\left[h_1 - (p + h_1 + h_2)\mathsf{P}\{B_0^{(1)} > 0\}\right] \\ &= (h_1 + h_2) - (p + h_1 + h_2)\mathsf{P}\{B_0 > 0\} - \mathsf{P}\{B_1 = 0\}g_1(y_1). \quad \square \end{aligned}$$

Things bring us at the point to derive *newsboy equations* for the optimal base-stock levels S_1 and S_2. Suppose that the demand distribution function F is continuous on $(0,\infty)$, and that there is no probability mass in 0, i.e., $F(0)=0$. Then $g_1(y_1)$ is a continuous function, and as an optimal base-stock level is a minimal point of $G_1(y_1)$, S_1 will be a zero point of $g_1(y_1)$, i.e., $g_1(S_1)=0$. This leads immediately to a newsboy equation for S_1; see Part (i) of Theorem 3. Next, by Lemma 4,

$$g_2(S_1,y_2)=(h_1+h_2)-(p+h_1+h_2)\mathsf{P}\{B_0>0\},\quad y_2\in\mathbb{R},$$

where B_0 is given by (9)–(10) with y_1 replaced by S_1. One can easily verify that this function is continuous as a function of y_2. Because S_2 is a minimizing point of $G_2(S_1,y_2)$, it will be a zero point of $g_2(S_1,y_2)$, i.e., $g_2(S_1,S_2)=0$. This leads immediately to a newsboy equation for S_2; see Part (ii) of the following theorem. The equation for S_2 is called a newsboy equation because it constitutes a generalization of the well-known newsboy equation for a single-stage system. Theorem 3 is stated to hold for a continuous demand distribution F, but, in fact, it holds if both $g_1(y_1)$ and $g_2(S_1,y_2)$ has a zero point.

Theorem 3 (cf. van Houtum and Zijm [62], Section 4). *Newsboy equations for the optimal base-stock levels—Let the echelon cost functions $c_n(\cdot)$ be given by (1)–(2), and let F be continuous on $(0,\infty)$ with $F(0)=0$. Then*

(i) *The optimal base-stock level S_1 for echelon 1 is such that*

$$\mathsf{P}\{B_0^{(1)}=0\}=\frac{p+h_2}{p+h_1+h_2},$$

with

$$B_0^{(1)}=(D_{t_0+l_2,\,t_0+l_2+l_1}-S_1)^+.$$

(ii) *Under a given optimal base-stock level S_1 for echelon 1, the optimal base-stock level S_2 for echelon 2 is such that*

$$\mathsf{P}\{B_0=0\}=\frac{p}{p+h_1+h_2},$$

with

$$\begin{aligned}B_1&=(D_{t_0,\,t_0+l_2-1}-(S_2-S_1))^+,\\ B_0&=(B_1+D_{t_0+l_2,\,t_0+l_2+l_1}-S_1)^+.\end{aligned}$$

This theorem says that, when S_1 is determined, then it is pretended that stockpoint 2 can always deliver (i.e., the analysis is limited to subsystem 1) and the value for S_1 is chosen such that the no-stockout probability at stage 1 is equal to $(p+h_2)/(p+h_1+h_2)$. Next, when S_2 is determined, then the full system is considered, the base-stock level for echelon 1 is fixed at S_1, and the value for S_2 is chosen such that the no-stockout probability at the most downstream stage 1 is equal to $p/(p+h_1+h_2)$. With this S_2, the demand over a longer lead time has to be covered, but we are allowed to have a lower no-stockout probability in the full system than in subsystem 1.

Like for a single-stage system, our generalized newsboy equations show the effect of the ratios of the parameters for inventory holding and penalty costs on the optimal base-stock levels. In addition, they reveal how physical stock is positioned in the chain as a function of the way value is being built up in the chain. This is seen as follows. The echelon holding cost parameters h_1 and h_2 are, in general, proportional to the values added at stages 1 and 2, respectively. W.l.o.g., we may norm the total added value such that $h_1+h_2=1$. In that case, h_n, $n=1,2$, is equal to the fraction of the added value in stage n over the total added value in the chain. Let us look at the values for S_1 and S_2 as a function of h_2, i.e., the fraction of added value at stage 2. The larger h_2, the closer $(p+h_2)/(p+h_1+h_2)=(p+h_2)/(p+1)$ comes to 1, and, thus, the larger S_1. The point S_2 is such that we have a no-stockout probability $p/(p+h_1+h_2)=p/(p+1)$ for the full system. This fraction is independent of h_2.

As S_1 is increasing as a function of h_2, S_2 will be decreasing (a larger S_1 implies that a slightly smaller value for S_2 is sufficient to obtain that $\mathsf{P}\{B_0 = 0\} = p/(p+1)$), and, thus, the difference $S_2 - S_1$ is decreasing as well. The average physical stock in stockpoint 2 at the end of a period equals $\mathsf{E}\{((S_2 - S_1) - D_{t_0,\, t_0+l_2-1})^+\}$ (cf. (11)) and is also decreasing as a function of h_2. The average physical stock in stockpoint 1 is likely to be increasing (because of the increased S_1 and only slightly decreased S_2; however, we have no proof for this property). In the extreme case that $h_2 = 1$, and thus $h_1 = 0$, there is no added value at all at stage 1. Then we may choose $S_1 = \infty$, in which case there is no safety stock held in stockpoint 2. This property holds in general when $h_1 = 0$.

Corollary 1. *There exists an optimal base-stock policy under which no safety stock is held in stockpoint* 2 *in case* $h_1 = 0$.

Proof. Suppose that $h_1 = 0$. Then, by Part (i) of Theorem 3, S_1 may be chosen equal to $S_1 = \infty$. This implies that, in each period, all goods arriving in stockpoint 2 are immediately forwarded to stockpoint 1, and, thus, there is never stock present in stockpoint 2 at the end of a period. □

3. Multiechelon, Serial Systems, and Computational Procedures

The whole analysis of §2 is easily generalized to serial systems with $N \geq 2$ stages. For the generalization of the optimality of base-stock policies and the decomposition result, see the remarks at the end of §2.2 (just before Remark 1). In this section, we present the cost formulas and newsboy equations as obtained for the N-stage system under linear inventory holding and penalty costs; see §3.1. After that, in §3.2, we describe both exact and efficient approximate computational procedures for the optimal base-stock levels and optimal costs.

3.1. Analytical Results

We first describe our model for the multiechelon, serial system, and introduce additional notation. We make the same assumptions as in §2, however, we now have N (≥ 2) stages, which are numbered from downstream to upstream as stages $1, 2, \ldots, N$. Periods are numbered $0, 1, \ldots$ Lead times are deterministic, and the lead time for stage n is denoted by l_n. The cumulative lead time for stages i, $n \leq i \leq N$, together is denoted by L_n; $L_n = \sum_{i=n}^{N} l_i$, and, for notational convenience, $L_{N+1} := 0$. The cumulative demand over periods $t_1, \ldots, t_2$, $0 \leq t_1 \leq t_2$, is denoted by $D_{t_1,\, t_2}$, F is the generic distribution function for one-period demand, and μ denotes the mean demand per period.

For the costs, we assume linear inventory holding and penalty costs. A cost of H_n, $n = 2, \ldots, N$, is charged for each unit that is in stock in stockpoint n at the end of a period and for each unit in the pipeline from the nth to the $(n-1)$th stockpoint. A cost of H_1 is charged for each unit that is in stock in stockpoint 1 at the end of a period, and a penalty $p > 0$ is charged per unit of backlog at stockpoint 1 at the end of a period. We assume that $H_1 \geq H_2 \geq \cdots \geq H_N \geq 0$; for notational convenience, $H_{N+1} = 0$. Next, we define $h_n := H_n - H_{n+1}$, $n = 1, \ldots, N$, as the additional inventory holding-cost parameters. Notice that $h_n \geq 0$ for all n. Under this cost structure and given levels x_n for the echelon stocks at the end of a period, the total inventory holding and backordering costs at the end of that period are equal to $\sum_{n=1}^{N} c_n(x_n)$, where $c_n(x_n)$ denotes the costs attached to echelon n (cf. (1)–(2) for $N = 2$):

$$c_1(x_1) = h_1 x_1 + (p + H_1) x_1^-,$$

$$c_n(x_n) = h_n x_n, \quad 2 \leq n \leq N.$$

Optimal base-stock levels follow from the minimization of average costs of a base-stock policy in subsystems. Subsystem n, $n = 1, \ldots, N$, is defined as the system consisting of the stages $1, \ldots, n$, and with infinite supply at stage $n+1$ (= external supplier of raw materials

in case $n = N$). As costs we have the echelon cost functions $c_i(\cdot)$ for the echelons $i = 1, \ldots, n$. Notice that subsystem N is identical to the full system. A base-stock policy for subsystem n is denoted by $(y_1, \ldots, y_n)$, with $y_i \in \mathbb{R}$ for all $i = 1, \ldots, n$, and the corresponding average costs are denoted by $G_n(y_1, \ldots, y_n)$. For this function, a similar expression may be derived as for the average costs of a two-echelon system in Theorem 2. We define $B_i^{(n)}$ as the shortfall as faced by stockpoint i, $1 \le i \le n$, and $B_0^{(n)}$ as the backlog at the end of an arbitrary period. For these variables, one easily derives similar recursive expressions as in (9)–(10). This leads directly to the following theorem.

Theorem 4 (cf. van Houtum and Zijm [62], van Houtum et al. [65]). *Let $1 \le n \le N$. For subsystem n, the average costs of a base-stock policy $(y_1, \ldots, y_n)$, with $y_i \in \mathbb{R}$ for all $i = 1, \ldots, n$, are equal to*

$$G_n(y_1, \ldots, y_n) = \sum_{i=1}^{n} h_i\big(y_i - \mathsf{E}B_i^{(n)} - (l_i + 1)\mu\big) + (p + H_1)\mathsf{E}B_0^{(n)},$$

with

$$B_n^{(n)} = 0, \tag{16}$$

$$B_i^{(n)} = \big(B_{i+1}^{(n)} + D_{t_0+L_{i+2},\, t_0+L_{i+1}-1} - (y_{i+1} - y_i)\big)^+, \quad 1 \le i \le n-1, \tag{17}$$

$$B_0^{(n)} = \big(B_1^{(n)} + D_{t_0+L_2,\, t_0+L_1} - y_1\big)^+ \tag{18}$$

(the equation for $B_i^{(n)}$, $1 \le i \le n-1$, vanishes in case $n = 1$).

An optimal base-stock level S_1 for stage 1 is obtained as a minimizer of the convex function $G_1(y_1)$. Next, under a given S_1, an optimal base-stock level S_2 for stage 2 is obtained as a minimizer of the function $G_2(S_1, y_2)$, which is known to be convex as a function of y_2; and so on. The optimal base-stock levels may also be obtained from partial derivatives. Define

$$g_n(y_1, \ldots, y_n) := \frac{\delta}{\delta y_n}\{G_n(y_1, \ldots, y_{n-1}, y_n)\}, \quad 1 \le n \le N,\ y_i \in \mathbb{R} \text{ for all } i = 1, \ldots, n.$$

Similar to Lemma 4, one can derive that

$$g_n(y_1, \ldots, y_n) = \sum_{i=1}^{n} h_i - (p + H_1)\mathsf{P}\big\{B_0^{(n)} > 0\big\} - \sum_{i=1}^{n-1} \mathsf{P}\big\{B_i^{(n)} = 0\big\} g_i(y_1, \ldots, y_i), \tag{19}$$

where the $B_i^{(n)}$ are given by (16)–(18) (in this formula the last sum vanishes in case $n = 1$). Under a continuous demand distribution F, $g_1(y_1)$ has a zero point, $g_2(S_1, y_2)$ has a point S_2 such that $g_2(S_1, S_2) = 0$, and so on. Then the last sum in (19) becomes equal to 0, and we get the following newsboy equations.

Theorem 5 (cf. van Houtum and Zijm [62], Theorem 5.1). *Newsboy equations for the optimal base-stock levels—Let F be continuous on $(0, \infty)$ with $F(0) = 0$. For $n = 1, 2, \ldots, N$, under given optimal base-stock levels $S_1, \ldots, S_{n-1}$ for the stages $1, \ldots, n-1$, S_n is such that*

$$\mathsf{P}\big\{B_0^{(n)} = 0\big\} = \frac{p + H_{n+1}}{p + H_1},$$

where $B_0^{(n)}$ is given by the recursive formulas (16)–(18) with y_i replaced by S_i for all i.

3.2. Computational Procedures

In case of a continuous demand distribution F with $F(0) = 0$, an optimal base-stock policy $(S_1, \ldots, S_N)$ and the corresponding average costs can be determined as follows. First, for $n = 1, \ldots, N$, S_n may be determined by the newsboy equation in Theorem 5. In general,

this newsboy equation cannot be solved analytically. Computational procedures can be developed, however. Suppose one has a computational procedure to compute $\mathsf{P}\{B_0^{(n)} = 0\}$ for a given arbitrary S_n. Then, an S_n that solves the newsboy equation is easily computed via bisection search. Once optimal base-stock levels have been determined for all stages, the optimal average costs $G_N(S_1, \ldots, S_N)$ follow from Theorem 4. Here, one needs a method to obtain the expected values of the $B_i^{(N)}$, $0 \le i \le N$. For both the computation of the optimal base-stock levels and the corresponding optimal costs, it suffices if one is able to evaluate the shortfalls/backlogs $B_i^{(n)}$ as given by (16)–(18). That is what we focus on in the rest of this subsection.

The shortfalls/backlogs $B_i^{(n)}$ may be determined recursively after a sufficiently fine discretization of the one-period demand distribution F. This is a first method. However, this method will be computationally inefficient in many cases, in particular, as N grows large. Therefore alternative procedures are desired. In §3.2.1, we describe an efficient, exact procedure for mixed Erlang demand, i.e., for the case that the one-period demand is a mixture of Erlang distributions with the same scale parameter. Such mixtures are relevant because the class of these mixtures is dense in the class of all distributions on $[0, \infty)$ (cf. Schassberger [53]). In §3.2.2, we describe a procedure based on two-moment fits. This is a fast, approximate procedure that is known to be accurate.

If the demand distribution F is *not* continuous, then Theorem 5 does not apply anymore, but Equation (19) still does. An optimal base-stock level for stage n is then found at the first point S_n where $g_n(S_1, \ldots, S_{n-1}, S_n) \ge 0$. Similar computations apply as described above, and the same methods may be used for the computation of the shortfalls/backlogs $B_i^{(n)}$. Via discretization, one still obtains an exact approach. The method of §3.2.2 is also applicable without further changes. The method of §3.2.1 may be applied after a (two-moment) fit of a mixed Erlang distribution on the one-period demand. That step is an approximate step, and for the rest the method is exact. A special case of noncontinuous demand is obtained in the case of a discrete product. Then, the demand distribution F is discrete as well, and base-stock and inventory levels may be limited to discrete values—in which case, Theorem 4 and Equation (19) are still valid. In this case, a direct recursive computation of the distributions of the shortfalls/backlogs $B_i^{(n)}$ may be efficient. For further details on this discrete product case, see §4.2.

3.2.1. Exact Procedure for Mixed Erlang Demands. The exact procedure as described here stems from van Houtum et al. [66], where for a generalized system with fixed replenishment intervals per stage, evaluation of shortfalls/backlogs of the same form as in (16)–(18) is needed. This procedure is closely related to the exact procedure described in van Houtum and Zijm [63], but the procedure as described here leads to simpler formulas and is easier to implement. The key idea behind the procedure is that we define a class of mixed Erlang distributions that is closed under the two basic operations in the expressions for the shortfalls/backlogs: Convolution and the so-called truncated shift.

Let us first define the class of mixed Erlang distributions that we use. We take $\lambda > 0$ as a given, and define a class of mixed Erlang random variables $\mathcal{C}_\lambda$. Let $X_{k,\lambda}$ be an Erlang distribution with $k \in \mathbb{N}_0$ phases and scale parameter λ. $X_{k,\lambda}$ may be interpreted as the sum of k independent, exponentially distributed random variables with parameter λ. Notice that we allow that $k = 0$. The distribution function of $X_{k,\lambda}$ is denoted by $E_{k,\lambda}$. For $k \in \mathbb{N}_0$,

$$E_{k,\lambda}(x) = 1 - \sum_{j=0}^{k-1} \frac{(\lambda x)^j}{j!} e^{-\lambda x}, \quad x \ge 0,$$

and $E_{k,\lambda}(x) = 0$ for all $x < 0$ (the sum $\sum_{j=0}^{k-1}$ is empty for $k = 0$). Let X be a pure mixture of the random variables $X_{k,\lambda}$, described by a discrete distribution $\{q_k\}_{k \in \mathbb{N}_0}$ on $\mathbb{N}_0$; i.e., $X = X_{k,\lambda}$ with probability q_k for all $k \in \mathbb{N}_0$. The distribution function of X is given by

$F_X(x) = \sum_{k=0}^{\infty} q_k E_{k,\lambda}(x),\ x \in \mathbb{R}$. Finally, we define random variable Y as the sum of a deterministic variable $d \geq 0$ and a pure mixture X; i.e., $Y = d + X$, and its distribution function is given by $F_Y(x) = \mathsf{P}\{d + X \leq x\} = F_X(x-d)$, $x \in \mathbb{R}$; this distribution is obtained by a shift of F_X to the right over a distance d. The class $\mathcal{C}_\lambda$ consists of all Ys that can be constructed in this way. Each $Y \in \mathcal{C}_\lambda$ is uniquely determined by a $d \geq 0$ and a discrete distribution $\{q_k\}_{k\in\mathbb{N}_0}$.

The first operation that we recognize in (16)–(18) is a convolution; i.e., $B_{i+1}^{(n)} + D_{t_0+L_{i+2},\,t_0+L_{i+1}-1}$ is a convolution of the random variables $B_{i+1}^{(n)}$ and $D_{t_0+L_{i+2},\,t_0+L_{i+1}-1}$, and $D_{t_0+L_{i+2},\,t_0+L_{i+1}-1}$ itself is a convolution of l_{i+1} one-period demands; and similarly for $B_1^{(n)} + D_{t_0+L_2,\,t_0+L_1}$. Let $Y \in \mathcal{C}_\lambda$ with parameters d and $\{q_k\}_{k\in\mathbb{N}_0}$, $\tilde{Y} \in \mathcal{C}_\lambda$ with parameters $\tilde{d}$ and $\{\tilde{q}_k\}_{k\in\mathbb{N}_0}$, and $\widehat{Y} := Y + \tilde{Y}$. Then, the sum $\widehat{Y}$ may be written as $\widehat{Y} = \hat{d} + \widehat{X}$, where $\hat{d} = d + \tilde{d}$ and $\widehat{X} = X + \tilde{X}$. Here, X is the pure mixture of Erlangs with discrete distribution $\{q_k\}_{k\in\mathbb{N}_0}$, and $\tilde{X}$ is the pure mixture given by $\{\tilde{q}_k\}_{k\in\mathbb{N}_0}$. It is easily seen that $\widehat{X}$ is also a pure mixture of Erlangs; its distribution $\{\hat{q}_k\}_{k\in\mathbb{N}_0}$ is obtained via the convolution of $\{q_k\}_{k\in\mathbb{N}_0}$ and $\{\tilde{q}_k\}_{k\in\mathbb{N}_0}$: $\hat{q}_k = \sum_{j=0}^{k} q_{k-j}\tilde{q}_j$, $k \in \mathbb{N}_0$. Hence, $\widehat{Y} \in \mathcal{C}_\lambda$. So, $\mathcal{C}_\lambda$ is closed under convolutions, and we have expressions to compute the parameters of an element that is obtained via a convolution.

The second operation that we recognize in (16)–(18) is a so-called *truncated shift*. Let Y be an arbitrary random variable (i.e., not necessarily an element of $\mathcal{C}_\lambda$), $a \in \mathbb{R}$, and $\widehat{Y} := (Y-a)^+$. If $a \leq 0$, then $\widehat{Y} = (-a) + Y$, and, thus, the distribution of $\widehat{Y}$ is obtained by a shift to the right of the distribution of Y over a distance $-a$. If $a > 0$, then the distribution of $\widehat{Y}$ is obtained by a shift to the left of the distribution of Y over a distance a, where the probability mass that would arrive in the negative range is absorbed in 0. Therefore, $\widehat{Y}$ is said to be a truncated shift of Y. Suppose, now that $Y \in \mathcal{C}_\lambda$ with parameters d and $\{q_k\}_{k\in\mathbb{N}_0}$, let $a \in \mathbb{R}$, and define $\widehat{Y} := (Y-a)^+$. Let X be the pure mixture of Erlangs given by $\{q_k\}_{k\in\mathbb{N}_0}$ (so, $Y = d + X$). We distinguish two cases: $a \leq d$ and $a > d$. If $a \leq d$, then $\widehat{Y} = (Y-a)^+ = (d + X - a)^+ = (d-a) + X$, and, thus, $\widehat{Y} \in \mathcal{C}_\lambda$ with parameters $d-a$ and $\{q_k\}_{k\in\mathbb{N}_0}$. Suppose now that $a > d$. Then

$$\widehat{Y} = (X - (a-d))^+ = (X_{k,\lambda} - (a-d))^+ \quad \text{with probability } q_k,\ k \in \mathbb{N}_0. \tag{20}$$

For each $k \in \mathbb{N}_0$, the k phases of $X_{k,\lambda}$ are equivalent to the first k interarrival times of a Poisson process with parameter λ, and $(X_{k,\lambda} - (a-d))^+$ depends on how many interarrival times have been completed at time instant $a-d$. With probability $[(\lambda(a-d))^j/j!]e^{-\lambda(a-d)}$, j phases of the Poisson process have been completed at time $a-d$, $j \in \mathbb{N}_0$. If $j < k$ phases have been completed, then there still are $k-j$ phases to go at time instant $a-d$, and, thus, then $(X_{k,\lambda} - (a-d))^+ = X_{k-j,\lambda}$. If $j \geq k$, then no phases are left, and $(X_{k,\lambda} - (a-d))^+ = 0$. Hence

$$(X_{k,\lambda} - (a-d))^+ = \begin{cases} X_{j,\lambda} & \text{with prob. } r_{k,j} = \dfrac{(\lambda(a-d))^{k-j}}{(k-j)!}e^{-\lambda(a-d)},\ j = 1,\ldots,k; \\ 0 & \text{with prob. } r_{k,0} = 1 - \displaystyle\sum_{j=0}^{k-1} \frac{(\lambda(a-d)^j}{j!}e^{-\lambda(a-d)}. \end{cases} \tag{21}$$

Combining this result and (20) shows that

$$\widehat{Y} = X_{j,\lambda} \quad \text{with probability } \hat{q}_j = \sum_{k=j}^{\infty} q_k r_{k,j},\ j \in \mathbb{N}_0.$$

As we see, $\widehat{Y}$ is a pure mixture of Erlangs in this case. This implies that $\widehat{Y} \in \mathcal{C}_\lambda$. So, $\mathcal{C}_\lambda$ is also closed under truncated shifts, and we have expressions to compute the parameters of an element that is obtained via a truncated shift.

Suppose now that the one-period demand D_{t_0,t_0} belongs to $\mathcal{C}_\lambda$ for some $\lambda > 0$; i.e., that $F = \sum_{k=0}^{\infty} q_k E_{k,\lambda}(x-d)$, $x \in \mathbb{R}$, where d is a nonnegative, real-valued constant and $\{q_k\}_{k\in\mathbb{N}_0}$ is a discrete distribution on $\mathbb{N}_0$. To obtain a continuous F with $F(0) = 0$, we require that $q_0 = 0$. Then each of the demand variables $D_{t_0+L_{i+2},t_0+L_{i+1}-1}$ and $D_{t_0+L_2,t_0+L_1}$ in (17)–(18) belongs to $\mathcal{C}_\lambda$ because they are convolutions of one-period demands. The shortfall $B_n^{(n)}$ in (18) is equal to $X_{0,\lambda}$ (and, thus, belongs to $\mathcal{C}_\lambda$). Next, for each $i = n-1, n-2, \ldots, 1$, the distribution of $B_i^{(n)}$ is obtained via a convolution, leading to the distribution of $B_{i+1}^{(n)} + D_{t_0+L_{i+2},t_0+L_{i+1}-1}$, followed by a truncated shift. Finally, $B_0^{(n)}$ is obtained via a convolution, leading to the distribution of $B_1^{(n)} + D_{t_0+L_2,t_0+L_1}$, followed by a truncated shift. In addition to these computations, it is simple to obtain the no-stockout probability $\mathsf{P}\{B_0^{(n)} = 0\}$ and/or expected values of the shortfalls/backlogs.

This completes the description of the exact computational procedure for the mixed Erlang demand case. Such a mixture is assumed to be given for this procedure. In practice, however, often only the first two moments of the one-period demand are given, and then a two-moment fit may be applied first: A so-called Erlang$(k-1,k)$ distribution can be fitted if the coefficient of variation of the demand is smaller than or equal to one, and a so-called Erlang$(1,k)$ distribution otherwise (these fits are further explained in §3.2.2). In principle, more moments may be fitted as desired, yielding a larger mixture.

The more general class of *phase-type distributions* is likewise closed under convolutions and truncated shifts. So, an exact procedure can also be derived for phase-type distributions, although computations become much more complicated.

Finally, it is relevant to note that the shortfalls/backlogs $B_i^{(n)}$ are equivalent to waiting times in a so-called *appointment system* (Vanden Bosch and Dietz [67], Wang [68]). Suppose you have a single server in which $n+1$ customers arrive. The customers are numbered $n, n-1, \ldots, 1, 0$, and they arrive at predetermined arrival times $0, y_n - y_{n-1}, \ldots, y_2 - y_1, y_1$. The service times for the customers $n, n-1, \ldots, 2, 1$ are given by the random variables $D_{t_0+L_{n+1},t_0+L_n-1}, D_{t_0+L_n,t_0+L_{n-1}-1}, \ldots, D_{t_0+L_3,t_0+L_2-1}, D_{t_0+L_2,t_0+L_1}$. Then, $B_i^{(n)}$ is the waiting time of customer i, $0 \le i \le n$ (cf. van Houtum and Zijm [63]). In fact, the exact procedure of this section may also be applied for the evaluation of waiting times in an appointment system if all service times belong to $\mathcal{C}_\lambda$ for a given $\lambda > 0$. The shortfalls/backlogs $B_i^{(n)}$ are also equivalent to waiting times in a multistage serial production system with planned lead times. For those systems, even a similar structure for the optimal policy and a decomposition result for the optimal planned lead times is obtained; see Gong et al. [38].

3.2.2. Efficient, Approximate Procedure Based on Two-Moment Fits. If one is satisfied with accurate approximations, then one may use the simple approximate procedure based on two-moment fits as described and tested in van Houtum and Zijm [62].

A two-moment fit may be applied to any nonnegative random variable X as follows. Let its mean μ_X (> 0) and coefficient of variation c_X (> 0) be given. Then, a mixture of two Erlangs may be fitted on X such that this mixture has the same first two moments as X (i.e., also the mean and coefficient of variation of this mixture are equal to μ_X and c_X, respectively). Let this mixture be denoted by $\widehat{X}$. Then, $\widehat{X} = X_{k_1,\lambda_1}$ with probability q_1 and $\widehat{X} = X_{k_2,\lambda_2}$ with probability $q_2 = 1 - q_1$.

The type of mixture that may be fitted on X depends on the value of c_X. We give three types of mixtures as described by Tijms [60]. If $c_X \le 1$, then we may fit an Erlang$(k-1,k)$ distribution, in which case, $k_1 = k-1$ and $k_2 = k$ for some $k \ge 2$ and $\lambda_1 = \lambda_2 = \lambda$. The Erlang$(k-1,k)$ distribution is a mixture of two Erlang distributions with the same scale parameter. The $k \ge 2$ is chosen such that $1/k < c_X^2 \le 1/(k-1)$. Next, q_1 and λ are taken equal to

$$q_1 = \frac{1}{1+c_X^2}\left[kc_X^2 - \sqrt{k(1+c_X^2) - k^2 c_X^2}\right], \quad \lambda = \frac{k - q_1}{\mu_X}.$$

If $c_X \geq 1$, then we may fit a hyperexponential or an Erlang$(1,k)$ distribution. Which of these two distributions is used may depend on further information that is available on X, e.g., on the shape of its probability density function (see also Tijms [60]). A hyperexponential distribution is a mixture of two exponential distributions, i.e., $k_1 = k_2 = 1$. In this case, multiple choices for λ_1, λ_2, q_1 are possible, and one choice that works is given by

$$\lambda_1 = \frac{2}{\mu_X}\left(1 + \sqrt{\frac{c_X^2 - 1/2}{c_X^2 + 1}}\right), \quad \lambda_2 = \frac{4}{\mu_X} - \lambda_1, \quad q_1 = \frac{\lambda_1(\lambda_2\mu_X - 1)}{\lambda_2 - \lambda_1}.$$

An Erlang$(1,k)$ distribution is a mixture of an exponential distribution and an Erlang distribution with the same scale parameter. Then $k_1 = 1$ and $\lambda_1 = \lambda_2 = \lambda$. The k_2 is set as the smallest $k_2 \geq 3$ for which $(k_2^2 + 4)/(4k_2) \geq c_X^2$. Next, q_1 and λ are taken equal to

$$q_1 = \frac{2k_2c_X^2 + k_2 - 2 - \sqrt{k_2^2 + 4 - 4k_2c_X^2}}{2(k_2 - 1)(1 + c_X^2)}, \quad \lambda = \frac{q_1 + k_2(1 - q_1)}{\mu_X}.$$

To approximate the shortfalls/backlogs $B_i^{(n)}$ in (16)–(18), we take the following steps. First, we determine the first two moments of $B_n^{(n)} + D_{t_0+L_{n+1},\,t_0+L_n} = D_{t_0+L_{n+1},\,t_0+L_n}$, and we fit a mixture of two Erlangs on these first two moments. Given this fit, $B_{n-1}^{(n)}$ is a truncated shift of $D_{t_0+L_{n+1},\,t_0+L_n}$, and via the observations made in §3.2.1 (among others, Equation (21)), it is straightforward to obtain the first two moments of $B_{n-1}^{(n)}$. Next, the first two moments of $B_{n-1}^{(n)} + D_{t_0+L_n,\,t_0+L_{n-1}}$ can be determined, and a mixture of two Erlangs may be fitted on these first two moments. This process is continued until a mixed Erlang distribution is obtained for $B_1^{(n)} + D_{t_0+L_2,\,t_0+L_1}$. From that last fit, it is straightforward to determine $\mathsf{E}B_0^{(n)}$ or $\mathsf{P}\{B_0^{(n)} = 0\} = \mathsf{P}\{B_1^{(n)} + D_{t_0+L_2,\,t_0+L_1} \leq y_1\}$. (In this procedure, in case the two-moment fit is applied to a nonnegative random variable X that consists of a deterministic part $d > 0$ and a nonnegative variable $\widetilde{X}$, i.e., $X = d + \widetilde{X}$, one may consider to take this deterministic part explicitly into account; i.e., one can apply the fit on $\widetilde{X}$ instead of X.)

In van Houtum and Zijm [62], the optimal policy and optimal costs of a multiechelon, serial system have been computed by both the approximate method based on two-moment fits and an exact method that is equivalent to the method of §3.2.1. A test bed has been defined in which holding cost parameters, lead times, the standard deviation of one-period demand, and the number of stages were varied, and an Erlang$(k-1,k)$ distribution has been assumed for the one-period demand (so that the exact method is applicable). The approximate method has appeared to be very accurate. The approximate procedure had a relative accuracy of 1% for the optimal base-stock levels and a relative accuracy of 2% for the optimal costs.

In case a higher accuracy is desired, the approximate method may be further improved by applying fits on the first three or even more moments; for three-moment fits, see Osogami and Harchol-Balter [47]. In the discrete product case (see also §4.2), one can use two-moments fits of discrete distribution as developed by Adan et al. [1].

4. Exact Solutions for Serial and Assembly Systems

In this section, we describe several generalizations/extensions of the multiechelon, serial system for which the optimal solution is known. First, in §§4.1–4.4, we describe modeling variants that we can easily deal with: Continuous review (and time) instead of periodic review, a discrete instead of continuous product, discounted instead of average costs, and the case with a γ-service level constraint instead of backordering costs. After that, in §4.5, we discuss the reduction of general assembly systems to serial systems. Next, in §§4.6–4.7, we describe the main results for serial systems with two different forms of batching: A fixed batch size per stage and a fixed replenishment interval per stage. Finally, some other extensions are discussed in §4.8.

4.1. Continuous Review

In §§2 and 3, we have assumed periodic review, but there is (almost) a full equivalence between periodic-review and continuous-review multiechelon systems: see Chen [10], Chen and Zheng [12], and Gallego and Zipkin [35]. Here, we demonstrate that equivalence for the two-echelon, serial system of §2.

Suppose we have the same two-echelon system as in §2, but now with continuous time and continuous review, i.e., we consider a time interval $[0, \infty)$ and ordering decisions may be taken at any time instant $t \in [0, \infty)$. Demands are assumed to occur according to a compound Poisson process. Hence, the demand process is memoryless, which is similar to i.i.d. demands in the periodic-review case. The total demand in an interval $(t_1, t_2]$ is denoted by D_{t_1,t_2}. So, D_{t_1,t_2} denotes the demand over a time interval with length $t_2 - t_1$; this is slightly different from the periodic-review case, where D_{t_1,t_2} was used to denote total demand over the periods $t_1, \ldots, t_2$ and, thus, corresponds to a length $t_2 - t_1 + 1$. The lead times l_1 and l_2 for the stockpoints 1 and 2 may be arbitrary, positive, real-valued numbers. Finally, the echelon cost functions $c_1(x_1)$ and $c_2(x_2)$ are now assumed to be cost *rate* functions.

For the continuous-review system, we define a cycle for each time instant $t_0 \in [0, \infty)$ in a similar way as for the periodic-review case. We get a similar picture as in Figure 3, but now, decision 2 is taken at time instant t_0, and decision 1 is taken at time instant $t_0 + l_2$, where the level z_1 is limited from above by $z_2 - D_{t_0,\, t_0+l_2}$ (in the periodic-review case, z_1 was bounded from above by $z_2 - D_{t_0,\, t_0+l_2-1}$; the change in this expression is due to the change in the definition of demands D_{t_1,t_2}). Decision 2 directly affects the echelon 2 cost rate at time instant $t_0 + l_2$, and decision 1 directly affects the echelon 1 cost rate at time instant $t_0 + l_2 + l_1$. These costs are given by exactly the same formulas as in the periodic-review case, i.e., by (3) and (4), respectively (notice, however, that the interpretation of $D_{t_0,\, t_0+l_2}$ and $D_{t_0+l_2,\, t_0+l_2+l_1}$ is slightly different now).

Next, for each $t_0 \in [0, \infty)$, we define the same relaxed single-cycle problem as in the periodic-review case; the only difference is that in the definition of problem (RP(t_0)) the demand variable $D_{t_0,\, t_0+l_2-1}$ is replaced by $D_{t_0,\, t_0+l_2}$. This relaxed single-cycle problem is solved in the same way as before. Therefore, we again find that there is an optimal base-stock policy (S_1, S_2) for problem (RP(t_0)), and the optimal base-stock levels follow from the minimization of convex functions $G_1(y_1)$ and $G_2(S_1, y_2)$; these functions are defined by (6) and (7), with $D_{t_0,\, t_0+l_2-1}$ replaced by $D_{t_0,\, t_0+l_2}$ in (7). Subsequently, for the infinite-horizon problem, it is optimal to follow base-stock policy (S_1, S_2) at each time instant, and, thus, base-stock policy (S_1, S_2) is also optimal for that problem. Finally, under linear holding and penalty costs, we obtain the same formulas as in the periodic-review case, but with $D_{t_0,\, t_0+l_2-1}$ replaced by $D_{t_0,\, t_0+l_2}$ in Equation (9) for B_1. Theorem 2 is still valid, and the newsboy equations of Theorem 3 hold as long as zero points exist for the functions $g_1(y_1)$ and $g_2(S_1, y_2)$. As the demand process is a compound Poisson process, the distribution functions for $D_{t_0,\, t_0+l_2}$ and $D_{t_0+l_2,\, t_0+l_2+l_1}$ have a positive probability mass in zero and, thus, it is not guaranteed that zero points exist for $g_1(y_1)$ and $g_2(S_1, y_2)$. This last issue constitutes a minor difference between the continuous-review and the periodic-review case. For the rest, all results are essentially the same.

4.2. Discrete Product

In §§2 and 3, we assumed that ordered amounts and inventory levels are continuous variables, mainly because that smooths the analysis. This assumption is natural for a continuous product for which customers may demand any real-valued amount. Further, the assumption makes sense for a discrete product with a sufficiently high mean demand and customers that may demand any integer-valued amount. However, for a discrete product with a low mean demand, it is more logical to limit order sizes and inventory levels to integer values. The analysis and results for the two-echelon system then change as follows. All cost functions, such as $G_1(y_1)$ in (6) and $G_2(y_1, y_2)$ in (7), are limited to discrete domains $\mathbb{Z}$ and $\mathbb{Z}^2$,

respectively. All results in the Lemmas 1–4 and Theorems 1–2 are still valid, where now the discrete variant of the definition of a convex function has to be taken, and the optimal base-stock levels S_1 and S_2 are obtained by the minimization of one-dimensional functions on $\mathbb{Z}$: $S_1 = \arg\min_{y_1 \in \mathbb{Z}} G_1(y_1)$ and $S_2 = \arg\min_{y_2 \in \mathbb{Z}} G_2(S_1, y_2)$. The newsboy equations of Theorem 3, which hold under linear inventory holding and penalty costs, become newsboy inequalities in this case (cf. Doğru et al. [23]). An optimal base-stock level S_1 for echelon 1 is obtained at the lowest $y_1 \in \mathbb{Z}$ for which

$$\mathsf{P}\{B_0^{(1)} = 0\} \geq \frac{p + h_2}{p + h_1 + h_2},$$

with $B_0^{(1)} = (D_{t_0+l_2,\, t_0+l_2+l_1} - y_1)^+$. Define $\varepsilon(S_1)$ as the difference between the left and right side of this inequality at the point S_1. Then, $\varepsilon(S_1) \geq 0$ and, in general, $\varepsilon(S_1)$ will be small. Next, an optimal base-stock level S_2 for echelon 2 is obtained at the lowest $y_2 \in \mathbb{Z}$ for which

$$\mathsf{P}\{B_0 = 0\} \geq \frac{p}{p + h_1 + h_2} + \mathsf{P}\{B_1 = 0\}\varepsilon(S_1), \tag{22}$$

with

$$B_1 = (D_{t_0,\, t_0+l_2-1} - (y_2 - S_1))^+,$$
$$B_0 = (B_1 + D_{t_0+l_2,\, t_0+l_2+l_1} - S_1)^+.$$

The second term on the right side of (22) is nonnegative, and, thus, under the optimal base-stock policy (S_1, S_2), the no-stockout probability in the full system is at least equal to $p/(p + h_1 + h_2)$. (For the generalization of these newsboy inequalities to serial systems with two or more stages Doğru et al. [23].)

4.3. Discounted Costs

Instead of minimizing average costs, one may be interested in minimizing discounted costs with a discount factor β, $0 < \beta < 1$. In practice, using discounted costs becomes relevant if the total lead time of a multiechelon system is long. The analysis hardly changes under discounted costs, because we can show on the basis of the two-echelon system of §2. Cycles are defined in precisely the same way as under average costs. For the cycle costs C_{t_0}, however, the echelon 2 costs $C_{t_0+l_2,\,2}$ have to be multiplied by a factor β^{l_2}, and the echelon 1 costs $C_{t_0+l_2+l_1,\,1}$ by a factor $\beta^{l_2+l_1}$ as they are charged l_2 and $l_2 + l_1$ periods after period t_0. Equivalently, in the single-cycle analysis, one may replace the echelon cost functions $c_1(x_1)$ and $c_2(x_2)$ by the modified functions $\tilde{c}_1(x_1) = \beta^{l_2+l_1} c_1(x_1)$ and $\tilde{c}_2(x_2) = \beta^{l_2} c_2(x_2)$. Under the presence of the discount factor, all convexity properties remain valid, and, thus, all main results hold again. Base-stock policies are optimal again. Under linear inventory holding and penalty costs, again, newsboy equations are obtained. For the optimal base-stock level S_1, the newsboy equation in Theorem 3(i) is still valid. For the optimal base-stock level S_2, we obtain the same newsboy equation as in Theorem 3(ii), but with the newsboy fractile $p/(p + h_1 + h_2)$ replaced by $(p - h_2(1 - \beta_1^l)/\beta_1^l)/(p + h_1 + h_2)$. Hence, the presence of the discount factor β has no effect on S_1, and it has a decreasing effect on S_2 (this is due to the decreased importance of echelon 1 costs $C_{t_0+l_2+l_1,\,1}$ relative to echelon 2 costs $C_{t_0+l_2,\,2}$ in a cycle).

4.4. γ-Service-Level Constraint

As stated before, when analyzing multiechelon systems, often linear inventory holding and penalty costs are assumed for the costs. The penalty costs are the counterpart for inventory holding costs, and optimal policies find a balance between these two types of costs. As an alternative for the penalty costs, one may assume a target service level, and then the

objective is to minimize the inventory holding costs subject to a service-level constraint. Both types of models are related because models with penalty costs are Lagrange relaxations of models with penalty costs; see van Houtum and Zijm [64] for an exposition of this relationship. The penalty costs that we have assumed in §§2.3 and 3.1 are of the so-called γ-type, and, therefore, the results of these sections can be extended to models with a so-called γ-service-level constraint. This is described below.

Consider the multiechelon serial system of §3.1. We still assume linear inventory holding costs, but we assume a γ-service-level constraint (which is equivalent to an average backlog constraint) instead of the linear penalty costs. The γ-service level is also known as the *modified fill rate*, and is closely related to the regular fill rate (= β-service level). For high service levels (more precisely, as long as demand is very rarely backordered for more than one period), both measures are virtually identical. Let γ_0 be the target γ-service level. We make the additional assumption that the demand distribution F has a *connected support*, i.e., F is strictly increasing from 0 to 1 on an interval $[a,b)$, with $0 \le a < b$ (b is allowed to be ∞). Under a base-stock policy $(y_1,\ldots,y_N)$, the average backlog at the end of a period equals $\mathsf{E}B_0^{(N)}$ (see Theorem 4), and the γ-service level is equal to

$$\gamma(y_1,\ldots,y_N) = 1 - \frac{\mathsf{E}B_0^{(N)}}{\mu};$$

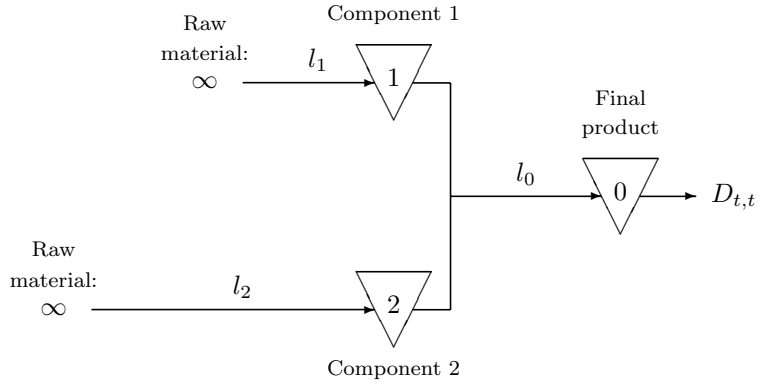
$B_0^{(N)}$ is given by (16)–(18). An optimal policy for the γ-service-level model is obtained as follows. First, if the optimal policy $(S_1,\ldots,S_N)$ for the penalty cost model (with penalty cost parameter p) has a γ-service-level $\gamma(S_1,\ldots,S_N) = \gamma(p) = \gamma_0$, then $(S_1,\ldots,S_N)$ is optimal for the service-level problem with target service-level γ_0 (cf. Everett [27]; see also Porteus [49] (Appendix B) and van Houtum and Zijm [64]). Second, the function $\gamma(p)$ is nondecreasing in p (cf. Everett [27], van Houtum and Zijm [64]). Third, under the assumption that F has connected support, one can show that the optimal base-stock levels $S_1,\ldots,S_N$ are continuous in p; thus, $\gamma(p)$ is also continuous in p. Moreover, $\gamma(p) \uparrow 1$ as $p \to \infty$. Therefore, the service-level problem with target $\gamma_0 < 1$ may be solved by repeatedly solving the penalty cost problem, tuning the penalty cost p until the γ-service-level $\gamma(p)$ of the optimal policy equals γ_0. This solves the service-level problem and implies that *the class of base-stock policies is also optimal for the service-level problem with a γ-service-level constraint.*

For a multiechelon model with a target β-service level (= fill rate) or a target α-service level (= no-stockout probability), the relaxed model is a model with β-type or α-type penalty costs, respectively. Then, the resulting echelon cost functions $c_n(x_n)$ are not convex anymore, and the approach of §2.2 does not work anymore to prove the optimality of base-stock policies. In fact, it is likely that the structure of optimal policies is more complicated for these types of service-level constraints. Nevertheless, it still may make sense to take the class of base-stock policies as a given and to optimize within this class, e.g., Boyaci and Gallego [6].

4.5. Assembly Systems

All results and insights presented so far are for serial multiechelon systems. They also apply to multiechelon systems with a pure assembly/convergent structure, in which each stage has one or more predecessors and uses input materials of all predecessors to assemble/produce one output product. This is due to Rosling [51], who showed that the analysis of pure assembly systems is equivalent to the analysis of a serial system (independently, in Langenhoff and Zijm [45], this result has been obtained for a two-echelon assembly system). We show this equivalence for a system in which a final product is obtained by the assembly of two components that are made from raw material; see Figure 5. The components are numbered 1 and 2 and the lead time for component $n = 1,2$ is $l_n \in \mathbb{N}$. The final product has index 0 and a lead time $l_0 \in \mathbb{N}_0$. W.l.o.g., we assume that one unit of the final product is made from one unit of Component 1 and one unit of Component 2. For the demand process, we have the same assumptions and notation as for the serial system in §2. For the costs,

FIGURE 5. An assembly system with two components.



we assume convex echelon cost functions $c_n(x_n)$, $n = 0, 1, 2$; in addition, we assume that $c_1(x_1)$ is nondecreasing.

If both components have equal lead times, i.e., if $l_1 = l_2$, then the two components may be replaced by one new virtual component of which each unit consists of one unit of Component 1 and one unit of Component 2, and the assembly system reduces to a two-echelon serial system.

From now on, we assume that the component lead times are different; w.l.o.g., assume that $l_2 > l_1$. For the resulting assembly system, an optimal policy may be derived along the same lines as in §2.2 for the two-echelon, serial system. At the beginning of each period $t_0 \in \mathbb{N}_0$, a single cycle starts and consists of the following three connected decisions:

- *Decision* 2: This decision concerns the order placed for Component 2 at the beginning of period t_0, by which echelon inventory position 2 is increased up to z_2. This decision leads to echelon 2 costs $c_2(z_2 - D_{t_0, t_0+l_2})$ at the end of period $t_0 + l_2$ and the resulting echelon stock 2 at the beginning of period $t_0 + l_2$ is $z_2 - D_{t_0, t_0+l_2-1}$.
- *Decision* 1: This decision concerns the order placed for component 1 at the beginning of period $t_0 + l_2 - l_1$, by which echelon inventory position 2 is increased up to z_1. This decision leads to echelon 1 costs $c_1(z_1 - D_{t_0+l_2-l_1, t_0+l_2})$ at the end of period $t_0 + l_2$, and the resulting echelon stock 1 at the beginning of period $t_0 + l_2$ is $z_1 - D_{t_0+l_2-l_1, t_0+l_2-1}$.
- *Decision* 0: This decision concerns the order placed for the final product at the beginning of period $t_0 + l_2$, by which echelon inventory position 0 is increased up to z_0. When this decision is taken, we are limited from above by the echelon stocks of the two components at that moment, i.e., $z_0 \le \min\{z_2 - D_{t_0, t_0+l_2-1}, z_1 - D_{t_0+l_2-l_1, t_0+l_2-1}\}$. Decision 0 leads to echelon 0 costs $c_0(z_0 - D_{t_0+l_2, t_0+l_2+l_0})$ at the end of period $t_0 + l_2 + l_0$.

We may now introduce an additional constraint based on the observation that it is never useful to order more for Component 1 than what is available in the parallel pipeline for Component 2. More precisely, the level z_1 to which echelon inventory position 1 is increased by decision 1 may be limited by the echelon stock 2 at that moment plus the amounts that will arrive at stockpoint 2 at the beginning of the periods $t_0 + l_2 - l_1, \ldots, t_0 + l_2$, i.e., by $z_2 - D_{t_0, t_0+l_2-l_1-1}$. If we take z_1 equal to $z_2 - D_{t_0, t_0+l_2-l_1-1}$, then echelon stock 2 and echelon stock 1 are both equal to $z_2 - D_{t_0, t_0+l_2-1}$ at the beginning of period $t_0 + l_2$. If we would take z_1 larger than $z_2 - D_{t_0, t_0+l_2-l_1-1}$, we would know beforehand that at the beginning of period $t_0 + l_2$, a portion of the arriving order at stockpoint 1 has to wait one or more periods for companion units in stockpoint 2. That portion would only lead to a larger echelon stock 1, and, thus, to equal or increased costs because $c_1(x_1)$ is nondecreasing. Hence, for decision 1, we introduce the additional constraint $z_1 \le z_2 - D_{t_0, t_0+l_2-l_1-1}$. As a result, the constraint for decision 0 simplifies to $z_0 \le z_1 - D_{t_0+l_2-l_1, t_0+l_2-1}$, and the decision structure for our assembly system becomes identical to the decision structure for a serial

system with three stages and lead times $l_0, l_1, l_2 - l_1$. Therefore, the optimal policy for our assembly system can be derived along the same lines as for that equivalent serial system (the cost structure in our assembly system is slightly different from the standard cost structure in a three-stage serial system, but it is still such that we have convex direct expected costs in the relaxed single-cycle problem). We again find that base-stock policies are optimal, and the optimal base-stock levels follow from the minimization of convex cost functions. In the special case of linear inventory holding and penalty costs, we obtain newsboy equations that are identical to the newsboy equations for a three-stage serial system with lead times $l_0, l_1, l_2 - l_1$, additional holding cost parameters h_0, h_1, h_2, and penalty cost parameter p.

The description above shows that the reduction of an assembly system to a serial system follows from a basic observation. Hence, this reduction is easily applied to many extensions of the Clark-Scarf system, among which the extensions in §§4.6–4.8.

4.6. Fixed Batch Size per Stage

In many supply chains, there may be setup times and costs involved each time that an order is placed. Setup costs may be modeled directly by fixed ordering costs. This leads to a serial system with a fixed ordering cost per stage, as studied by Clark and Scarf [15]. These fixed ordering costs cannot be captured by convex cost functions $c_n(x_n)$, and, thus, the analysis of §2 does not work anymore. In fact, an optimal solution seems to be complicated in this case; an exception is the case with a fixed ordering cost at the most upstream stage only (see also §4.8).

An alternative way to limit the number of orders per stage is by the introduction of a fixed batch size per stage, a fixed replenishment interval per stage, or a combination of both. These limitations may be determined at the first decision level of the hierarchical approach as discussed at the beginning of §1. In this subsection, we discuss the case with a fixed batch size per stage.

Consider the multiechelon, serial system as described in §3, and assume that a fixed batch size Q_n applies for stage n, $n = 1, \ldots, N$. This means that stage n is allowed to order at the beginning of each period, but the size of each order has to be an integer multiple of Q_n. There are no fixed ordering costs. The fixed batch size Q_{n+1} for stage $n+1$ is assumed to be an integer multiple of the fixed batch size for stage n, $n = 1, \ldots, N-1$. This is known as the integer-ratio constraint. This constraint facilitates the analysis and reflects that the further upstream we are in a supply chain, the higher the setup times and costs tend to be, and, thus, larger batch sizes are desired. We also assume that at time 0, the physical stock in stage n is an integer multiple of Q_{n-1}, $n = 2, \ldots, N$. For this system, Chen [10] (see also Chen [9]) derived the following optimal policy structure, via the approach that we used in §2.2. Each stage n, $n = 1, \ldots, N$, has to control its echelon inventory position by an (s, Q)-policy with fixed batch size Q_n and a reorder level s_n that follows from the minimization of a one-dimensional convex function. This policy is called a multiechelon (s, Q)-policy, and is a generalized form of a base-stock policy. Under a base-stock policy, each stage aims to bring its echelon inventory position back to the same point at the beginning of each period, while each stage aims to bring its echelon inventory position back to the interval $(s, s+Q]$ under a multiechelon (s, Q)-policy. For the case with linear inventory holding and penalty costs, Doğru et al. [24] generalized the cost formulas of Theorem 4 and the newsboy equations of Theorem 5, which now hold for the reorder levels s_n. In fact, for each $n = 1, \ldots, N$, the newsboy equation itself as given in Theorem 5 does not change; there are only a few changes in the recursive formulas (16)–(18) for the backlogs $B_0^{(n)}$.

4.7. Fixed Replenishment Interval per Stage

An alternative way to limit the number of orders per stage is by fixed replenishment intervals. Fixed replenishment intervals facilitate freight consolidations and logistics/production

scheduling and are, therefore, often observed in practice (cf. Graves [39]). In this subsection, we summarize the main results for such systems.

Consider the multiechelon, serial system as described in §3, and assume that a fixed replenishment interval T_n is specified for stage n, $n = 1, \ldots, N$. In this case, orders may have any size, but stage n is only allowed to order at the beginning of every T_n periods. The replenishment interval T_{n+1} of stage $n+1$ is assumed to be an integer multiple of the replenishment interval T_n of stage n, $n = 1, \ldots, N-1$ (integer-ratio constraint). In addition, we assume that the replenishment epochs are timed such that arriving materials at one stockpoint can be forwarded immediately to the next stockpoint if desired (synchronization constraint). This system has been analyzed in van Houtum et al. [66], along essentially the same lines as in §2.2. The main difference is constituted by the definition of a cycle. Consider, for example, a system with $N = 2$ stages. Then, a cycle is defined for each period t_0 in which stage 2 is allowed to order. An order by stage 2 in such a period t_0 directly affects the echelon 2 costs in the periods $t_0 + l_2, t_0 + l_2 + 1, \ldots, t_0 + l_2 + T_2 - 1$, and it limits the levels to which echelon inventory position 1 may be increased in the periods $t_0 + l_2, t_0 + l_2 + T_1, \ldots, t_0 + l_2 + kT_1$, where $k = T_2/T_1$. Further, each order by stage 1 in one of these periods $t' = t_0 + l_2, t_0 + l_2 + T_1, \ldots, t_0 + l_2 + kT_1$ has a direct effect on the echelon 1 costs in the periods $t' + l_1, t' + l_1 + 1, \ldots, t' + l_1 + T_1 - 1$. A cycle now consists of $k + 1$ decisions, one decision for stage 2 and k decisions for stage 1, and the cycle costs consist of the echelon 2 costs in the periods $t_0 + l_2, t_0 + l_2 + 1, \ldots, t_0 + l_2 + T_2 - 1$ and the echelon 1 costs in the periods $t_0 + l_2 + l_1, t_0 + l_2 + l_1 + 1, \ldots, t_0 + l_2 + l_1 + T_2 - 1$. Based on this definition of a cycle, all main results of the Clark-Scarf model have been generalized in van Houtum et al. [66]. In this case, we find a multiechelon (T, S)-policy as optimal policy; i.e., at the beginning of every T_n periods, stage n orders according to a base-stock policy with level S_n. For the newsboy equations, we now have to look at the average no-stockout probability over multiple periods, but we keep the same newsboy fractiles.

It is also possible to use both fixed batch sizes and fixed replenishment intervals. Serial systems with that combination have been analyzed by Chao and Zhou [8]. They combined the insights of Chen [10] and van Houtum et al. [66], and showed that the structure of the optimal policy is obtained by the combination of multiechelon (s, Q)- and (T, S)-policies.

For a cost comparison between serial systems with fixed batch sizes and serial systems with fixed replenishment intervals, we refer to Feng and Rao [32]. For a system with linear inventory holding costs, linear penalty costs, and fixed ordering costs, they compared the optimal multiechelon (T, S)-policy to the optimal multiechelon (s, Q)-policy. Multiechelon (s, Q)-policies lead to lower costs in general, but the differences in costs are relatively small. Hence, multiechelon (T, S)-policies are easily more attractive in situations in which freight consolidations and other coordination issues are important.

4.8. Other Extensions

There are a few more multiechelon, serial systems for which the structure of the optimal policy has been derived. Chen and Song [11] derived the optimal policy for a serial system with Markov-modulated demand, and Gallego and Özer [33] for a serial system with a specific form of advance demand information. In both cases, generalized forms of base-stock policies are optimal. Generalized base-stock policies may also be optimal for serial systems with an additional feature for the most upstream case.

Consider, for example, the two-echelon, serial system of §2 with a fixed capacity C for the upstream stage. Due to this fixed capacity, the upstream stage is never allowed to order more than C units in any period. For this system, a (modified) base-stock policy with parameters (S_1, S_2) is optimal (cf. Zijm and van Houtum [69]). This result is obtained as follows. Define cycles, cycle costs, and the relaxed single-cycle problem in a similar way as in §2.2. For the downstream stage of the relaxed single-cycle problem, one can show that a base-stock policy with a level S_1 is optimal. Next, one can conclude that it is optimal for stage 1 to follow

this base-stock policy in all periods. What remains is an infinite-horizon problem for stage 2 with a convex cost function $G_2(S_1, y_2)$ that denotes the costs attached to a period t_0 if the inventory position of echelon 2 in that period is increased to level y_2. This problem fits in the single-stage, capacitated inventory model as analyzed by Federgruen and Zipkin [30, 31]. Hence, for echelon 2, a so-called modified base-stock policy is optimal, i.e., at the beginning of each period, echelon 2 has to increase its echelon inventory position to a level S_2 if the fixed capacity allows this, and, otherwise, the echelon inventory position is increased as far as possible by an order of size C. The difference between S_2 and the actual level to which echelon inventory position 2 is increased is called a shortfall and its steady-state distribution is identical to the steady-state waiting time in an equivalent $D|G|1$ queue (cf. Tayur [58], Zijm and van Houtum [69]). By exploiting this observation, the results in Theorems 2 and 3 are easily generalized. For a multiechelon, serial system with a fixed capacity constraint at the most upstream stage, the optimality of base-stock policies is obtained in the same way.

Similarly, the optimal policy for a multiechelon, serial system with a fixed ordering cost for the upstream stage is obtained. In this case, all stages except the most upstream one has to follow a base-stock policy, and for the most upstream stage, it is optimal to follow an (s, S)-policy (cf. Clark and Scarf [15]). The policy for the most upstream stage follows from the fact that an (s, S)-policy is optimal for a single-stage inventory system with fixed ordering costs (cf. Scarf [52]).

Finally, Shang and Song [54] (see also Boyaci et al. [7]) obtained interesting results for the multiechelon, serial system by the definition of lower- and upper-bound subsystems for the subsystems $1, \ldots, N$ for the case with linear inventory holding and penalty costs. The upper-bound subsystems have a newsboy solution and have been shown to lead to lower bounds S_n^l for the optimal base-stock levels S_n. The lower-bound subsystems also have a newsboy solution and lead to upper bounds S_n^u for the optimal base-stock levels S_n. The weighted averages $(S_n^l + S_n^u)/2$ have appeared to be rather accurate approximations for the optimal base-stock levels S_n. An advantage of these approximations is that they are easy to compute. An alternative newsboy-type approximation has been developed by Gallego and Özer [34]. In Shang and Song [55], the bounds of Shang and Song [54] have been generalized to serial systems with a fixed batch size per stage; for a connection between these bounds and the newsboy equations for the optimal base-stock/reorder levels, see Doğru et al. [24].

5. Distribution and General Systems

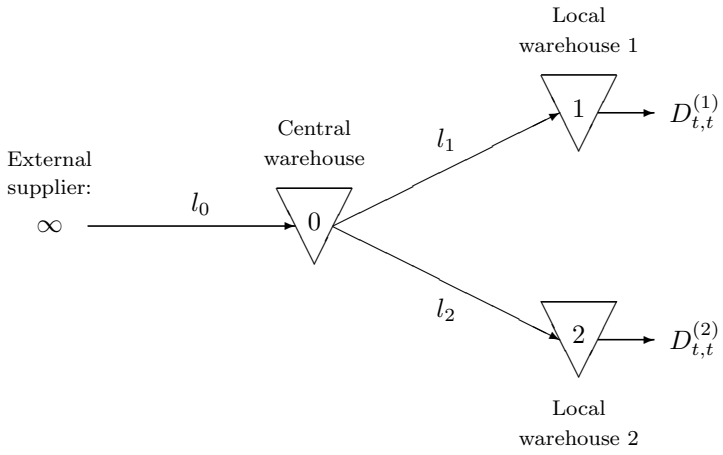
The theory presented in the previous sections shows that generalized base-stock policies are optimal for a variety of multiechelon systems with a pure serial or assembly/convergent structure, that optimal base-stock levels follow from the minimization of convex, one-dimensional functions, and that optimal base-stock levels satisfy newsboy equations for many systems. In §§5.1 and 5.2, we discuss systems with a pure distribution/divergent structure. Nice results may be derived for such systems under the so-called *balance assumption*. Without the balance assumption, however, the structure of the optimal policy may be complicated, and alternative approaches have to be taken in that case. In §5.3, we discuss general systems, with a mixed convergent-divergent structure. That are the systems that often occur in practice. We briefly discuss the approaches that have been developed for such systems.

5.1. A Basic Distribution System

In this subsection, we first extend the analysis of the two-echelon, serial system to a very basic distribution system. While doing that, we will introduce the balance assumption. As we shall see, the balance assumption, or, better, imbalance between inventories of different local stockpoints, is the key problem in the analysis of distribution systems.

Consider the distribution/divergent system depicted in Figure 6. In this system, there is one central stockpoint supplied by an external supplier, and two successive stockpoints

FIGURE 6. A two-echelon distribution system with two local warehouses.



supplied by this central stockpoint. Such a system may occur in a production environment, in which an intermediate product is used in two different final products. Alternatively, we obtain such a structure in a distribution network in which a product is kept on stock in a central warehouse and two different local warehouses. From now on, we use the terminology that is common for the latter environment.

For our distribution system, we make similar assumptions for the two-echelon, serial system of §2. The local warehouses are numbered 1 and 2, and we also denote them as stockpoints 1 and 2. The central warehouse is denoted as stockpoint 0. We have periods numbered $0, 1, \ldots$. The central warehouse has a deterministic lead time $l_0 \in \mathbb{N}$, and local warehouse n has a deterministic lead time $l_n \in \mathbb{N}_0$, $n = 1, 2$. Demands at local warehouse $n = 1, 2$ in different periods are independent and identically distributed on $[0, \infty)$, and the demands at one local warehouse are independent of the demands at the other local warehouse. The cumulative demand at local warehouse n over periods $t_1, \ldots, t_2$, $0 \le t_1 \le t_2$, is denoted by $D_{t_1,t_2}^{(n)}$, and the total demand at both warehouses together over those periods is denoted by $D_{t_1,t_2} = D_{t_1,t_2}^{(1)} + D_{t_1,t_2}^{(2)}$.

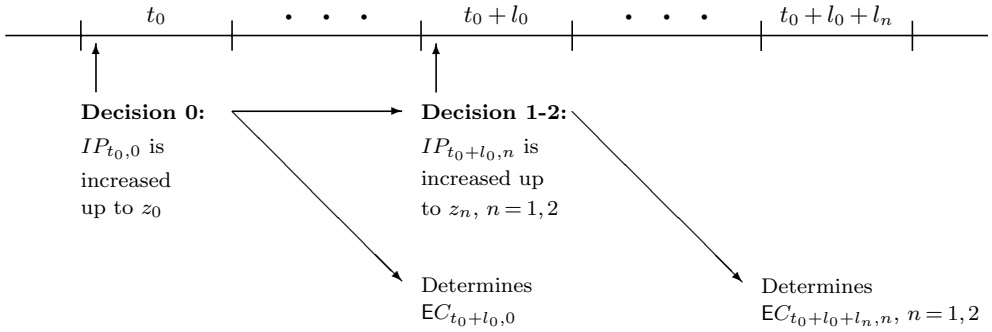
The costs are described by convex echelon cost functions $c_n(x_n)$. A special cost structure is constituted by linear inventory holding and penalty costs. Under that structure, a cost h_0 (≥ 0) is charged for each unit on stock in the central warehouse at the end of a period and for each unit in the pipelines from the central warehouse to the local warehouses. A cost $h_0 + h_n$ $(h_n \ge 0)$ is charged for each unit on stock in local warehouse n at the end of a period, and a penalty cost p_n is charged per unit of backordered demand at local warehouse n at the end of a period, $n = 1, 2$. Let x_n be echelon stock n at the end of a period. Then, the total inventory holding and penalty costs at the end of a period can be shown to be equal to $\sum_{n=0}^{2} c_n(x_n)$ with

$$c_0(x_0) = h_0 x_0,$$
$$c_n(x_n) = h_n x_n + (p_n + h_n + h_0) x_n^-, \quad n = 1, 2.$$

The objective is to minimize the average costs over the infinite horizon. We denote this problem as problem (P).

For the analysis, we follow the same steps as in §2.2. We start with the definition of cycles and cycle costs. Let $IL_{t,n}$ and $IP_{t,n}$ denote echelon stock n and echelon inventory position n at the beginning of period t (just before demand occurs), and let $C_{t,n}$ be the costs attached to echelon n at the end of period t. A cycle starts with an order placed by the central warehouse at the beginning of a period $t_0 \in \mathbb{N}_0$. This decision is called decision 0. By this decision, $IP_{t_0,0}$ becomes equal to some level z_0. First of all, this decision determines

FIGURE 7. The consequences of the decisions 0 and 1-2.



the echelon 0 costs at the end of period $t_0 + l_0$:

$$\mathsf{E}\{C_{t_0+l_0,0} | IP_{t_0,0} = z_0\} = \mathsf{E}\{c_0(z_0 - D_{t_0,t_0+l_0})\}.$$

Second, by this decision, echelon stock 0 at the beginning of period $t_0 + l_0$ becomes equal to $IL_{t_0+l_0,0} = z_0 - D_{t_0,t_0+l_0-1}$, and this directly limits the levels to which one can increase the echelon inventory positions $IP_{t_0+l_0,n}$, $n = 1, 2$, of the local warehouses at that moment. The latter decision is denoted as decision 1-2. Suppose that by this decision, $IP_{t_0+l_0,n}$ becomes equal to z_n, $n = 1, 2$. The sum $z_1 + z_2$ of these levels is limited from above by $z_0 - D_{t_0,t_0+l_0-1}$. Decision 1-2 directly affects the echelon n costs at the end of period $t_0 + l_0 + l_n$.

$$\mathsf{E}\{C_{t_0+l_0+l_n,n} | IP_{t_0+l_0,n} = z_n\} = \mathsf{E}\Big\{c_n\big(z_n - D^{(n)}_{t_0+l_0,t_0+l_0+l_n}\big)\Big\}, \quad n = 1, 2.$$

The cycle costs C_{t_0} are equal to $C_{t_0+l_0,0} + \sum_{n=1}^{2} C_{t_0+l_0+l_n,n}$. When the decisions 0 and 1-2 are taken, there is also bounding from below, but this is ignored for the moment. For a visualization of these decisions and the cycle costs; see Figure 7.

The second step of the analysis consists of the definition of the relaxed single-cycle problem. We obtain the following relaxed problem.

$$\begin{aligned}
(\mathrm{RP}(t_0)): \quad \text{Min} \quad & \mathsf{E}C_{t_0} = \mathsf{E}C_{t_0+l_0,0} + \sum_{n=1}^{2} \mathsf{E}C_{t_0+l_0+l_n,n} \\
\text{s.t.} \quad & \mathsf{E}C_{t_0+l_0,0} = \mathsf{E}\{c_0(z_0 - D_{t_0,t_0+l_0})\}, \\
& \mathsf{E}C_{t_0+l_0+l_n,n} = \mathsf{E}\Big\{c_n\big(z_n - D^{(n)}_{t_0+l_0,t_0+l_0+l_n}\big)\Big\}, \quad n = 1, 2, \\
& z_1 + z_2 \le IL_{t_0+l_0,0}, \\
& IL_{t_0+l_0,0} = z_0 - D_{t_0,t_0+l_0-1}.
\end{aligned}$$

Problem $(\mathrm{RP}(t_0))$ is a two-stage stochastic dynamic programming problem. Decision 0 is described by z_0 and is not limited at all. The resulting direct expected costs are equal to $\mathsf{E}\{c_0(z_0 - D_{t_0,t_0+l_0})\}$. Decision 1-2 is described by z_1 and z_2, and, via the constraint $z_1 + z_2 \le IL_{t_0+l_0,0}$, its decision space depends on $IL_{t_0+l_0,0}$. Hence, we use $IL_{t_0+l_0,0}$ to describe the state of the system when decision 1-2 is taken. This state depends on decision 2 via the relation $IL_{t_0+l_0,0} = z_0 - D_{t_0,t_0+l_0-1}$. Decision 1-2 results in direct expected costs $\sum_{n=1}^{2} \mathsf{E}\{c_n(z_n - D^{(n)}_{t_0+l_0,t_0+l_0+l_n})\}$.

We see decision 1-2 as a decision that decides on two issues simultaneously.

- the aggregate level $z_{loc} = z_1 + z_2$ to which the echelon inventory positions $IP_{t_0+l_0,n}$, $n = 1, 2$, together are increased; and
- the allocation of this total amount z_{loc} to the echelons 1 and 2, which is described by z_1 and z_2.

Let us first look at the allocation part. Suppose that a total amount $z_{loc} = x$, $x \in \mathbb{R}$, is being allocated. Then, z_1 and z_2 are obtained by the following allocation problem:

$$(\mathrm{AP}(x)): \quad \min \quad \sum_{n=1}^{2} \mathsf{E}\Big\{c_n\big(z_n - D^{(n)}_{t_0+l_0,\, t_0+l_0+l_n}\big)\Big\}$$
$$\text{s.t.} \quad z_1 + z_2 = x.$$

The optimal solution of problem $(\mathrm{AP}(x))$ is denoted by $z_n^*(x)$, $n = 1, 2$, and the optimal costs are denoted by $G_{loc}(x)$. The functions $z_n^*(x)$ are called *optimal allocation functions*. Because $c_n(\cdot)$ is convex, also $\mathsf{E}\{c_n(z_n - D^{(n)}_{t_0+l_0,\, t_0+l_0+l_n})\}$ is convex as a function of z_n, and one can show that the optimal costs $G_{loc}(x)$ are convex as a function of x. Let S_{loc} be a point in which $G_{loc}(x)$ is minimized; this point is such that $S_n = z_n^*(S_{loc})$ minimizes $\mathsf{E}\{c_n(z_n - D^{(n)}_{t_0+l_0,\, t_0+l_0+l_n})\}$, $n = 1, 2$ (we allow that S_{loc} and the S_n's are infinite). Decision 1-2 is taken optimally by increasing the sum of the echelon inventory positions $n = 1, 2$ to level $x = \min\{IL_{t_0+l_0,\,0}, S_{loc}\}$, i.e., according to a base-stock policy with level S_{loc}, and by allocating according to the optimal allocations $z_n^*(x)$, $n = 1, 2$.

Given the optimal solution for decision 1-2, we obtain total cycle costs

$$G_0(z_0) = \mathsf{E}\{c_0(z_0 - D_{t_0,\, t_0+l_0}) + G_{loc}(\min\{z_0 - D_{t_0,\, t_0+l_0-1}, S_{loc}\})\}$$

as a result of the level z_0 to which $IP_{t_0,\,0}$ is increased. Also, this function may be shown to be convex. Hence, for decision 0 it is optimal to follow a base-stock policy with level S_0, where S_0 is a minimizing point of $G_0(z_0)$. The optimal costs of problem $(\mathrm{RP}(t_0))$ are given by $G_0(S_0)$. Notice that the optimal policy for problem $(\mathrm{RP}(t_0))$ is described by the base-stock levels S_{loc} and S_0 and the optimal allocation functions $z_n^*(x)$, $n = 1, 2$.

We now arrive at the third step of the analysis. The optimal costs $G_0(S_0)$ constitute a lower bound LB for the optimal costs C_P of the infinite-horizon problem (P). Next, suppose that we apply the optimal policy of problem $(\mathrm{RP}(t_0))$ in each period of problem (P). Then, for echelon inventory position 0 and the sum of the echelon inventory positions $n = 1, 2$, we can follow base-stock policies with levels S_0 and S_{loc}, respectively; i.e., for these echelon inventory positions, the ordering behavior is precisely as in problem $(\mathrm{RP}(t_0))$. However, the allocation of the amount $x = \min\{IL_{t,\,0}, S_{loc}\}$ to echelons 1 and 2 at the beginning of period t may be problematic for some $t \in \mathbb{N}_0$. We would like to allocate $z_1^*(x)$ and $z_2^*(x)$, respectively, but it may happen that one level is below the current echelon inventory position. We demonstrate this by a possible sample path.

First, suppose that our distribution system is such that we have strictly increasing functions $z_n^*(x)$, $n = 1, 2$. Next, suppose that at the beginning of some period t, the echelon stock of the central warehouse is precisely equal to S_{loc}; i.e., $IL_{t,\,0} = S_{loc}$. Then, at the beginning of period t, the echelon inventory positions 1 and 2 are increased to levels $z_1^*(S_{loc}) = S_1$ and $z_2^*(S_{loc}) = S_2$, respectively, and no physical stock is left in the central warehouse. Next, suppose that in period t, zero demand occurred at local warehouse 1, and a positive demand d_2 occurs at local warehouse 2. Then, at the beginning of period $t+1$, the echelon inventory positions of echelons 1 and 2 before ordering are equal to $\widetilde{IP}_{t+1,\,1} = S_1$ and $\widetilde{IP}_{t+1,\,1} = S_2 - d_2$, respectively. Next, suppose that the order placed by the central warehouse in period $t - l_0 + 1$ was zero (because the total demand in period $t - l_0$ was zero), then nothing arrives in the central warehouse in period $t+1$ and, thus, $IL_{t+1,\,0} = S_{loc} - d_2$. We now would like to allocate $z_1^*(IL_{t+1,\,0})$ and $z_2^*(IL_{t+1,\,0})$ to echelons 1 and 2, respectively. However,

$$z_1^*(IL_{t+1,\,0}) < z_1^*(S_{loc}) = S_1 = \widetilde{IP}_{t+1,\,1},$$

i.e., echelon inventory position 1 before ordering is larger than the level to which echelon inventory position 1 should be increased according to the optimal policy for problem $(\mathrm{RP}(t_0))$. We say that there is *imbalance* between the echelon inventory positions 1 and 2.

Here, we described one situation that leads to imbalance. In general, it may occur if there is a big demand in one local warehouse, while there is a small demand in the other local warehouse, and not much stock is available at the central warehouse to balance the inventories again.

Because of a possible imbalance, the allocation cannot be executed according to the functions $z_n^*(x)$, $n = 1, 2$ in all periods. In the periods with imbalance, one can balance the echelon inventory positions as much as possible. If for local warehouse 1, the current inventory position is above the desired level according to the functions $z_n^*(x)$, then this is done by keeping echelon inventory position 1 at the current level and allocating the rest to echelon 2, and vice versa. This is known as myopic allocation. By following this rule, we obtain a feasible policy for problem (P) that leads to an upper bound UB for C_P; this UB may be determined via simulation. We call this policy the LB heuristic. The distance between UB and C_P denotes how well the LB heuristic performs. This distance $UB - C_P$, and also the distance $UB - LB$, will be small if imbalance occurs in relatively few periods only and if the imbalance is rather limited in those periods.

Clearly, due to the phenomenon of imbalance, the analysis of §2.2 for the two-echelon, serial system cannot be directly generalized to our basic distribution system. However, the generalization is possible if we assume that the echelon inventory positions $n = 1, 2$ are always balanced after allocation in all periods. This is equivalent to allowing that an echelon inventory position $n = 1, 2$ is decreased by the allocation, i.e., the corresponding local warehouse receives a negative shipment from the central warehouse. This assumption is called the *balance assumption*. Under the balance assumption, the optimal policy of problem (RP(t_0)) is also optimal for problem (P). This implies that then a base-stock policy, in combination with the optimal allocation functions $z_n^*(x)$, $n = 1, 2$, is optimal, and the optimal base-stock levels and the functions $z_n^*(x)$ can be determined sequentially (cf. Federgruen and Zipkin [28, 29]). The latter property generalizes the decomposition result. In addition, under linear inventory holding and penalty costs, the newsboy equations for the optimal base-stock levels can be generalized (Diks and de Kok [19], Doğru et al. [23]).

5.2. Literature on Distribution Systems

The research on distribution systems has a long history. Clark and Scarf [14] recognized already that base-stock policies are not optimal in general (i.e., without the balance assumption). Eppen and Schrage [25] introduced the balance assumption for a two-echelon, distribution system consisting of a stockless central warehouse and multiple local warehouses (they called that assumption the "allocation assumption"). For a two-echelon, distribution system with a stock-keeping central warehouse, the optimality of base-stock policies under the balance assumption and the decomposition result were derived by Federgruen and Zipkin [28, 29]. Diks and de Kok [19] extended these results to multiechelon, distribution systems. In this literature, linear inventory holding and penalty costs mainly were considered; it is straightforward to extend these results to general convex cost functions $c_n(\cdot)$. Under linear inventory holding and penalty costs, newsboy equations for the optimal base-stock levels have been derived for a general distribution system with continuous demand by Diks and de Kok [19] and for a two-echelon distribution system with discrete demand by Doğru et al. [23].

The above results give useful insights, however, the balance assumption is not always justified. Hence, it is relevant to know how well base-stock policies with optimal allocation functions perform if the balance assumption is not made, i.e., how well the LB heuristic as defined above performs. In Doğru et al. [22], the performance of the LB heuristic has been evaluated in a very large test bed of more than 5,000 instances for two-echelon distribution systems with symmetric and asymmetric local warehouses and with linear inventory holding and penalty costs. Notice that the optimal costs C_P can be determined by stochastic dynamic programming, but because of the curse of dimensionality, this is only possible for small-size instances with discrete demand. For that reason, $(UB - LB)/LB$ instead of $(UB - C_P)/C_P$

was used as the measure for the performance of the LB heuristic. It appeared that the LB heuristic performs well in many instances, but a large gap $(UB - LB)/LB$ may also easily occur, and even large gaps of more than 100% were found for some instances. Large gaps mainly occur if the demands at the local warehouses have high coefficients of variation, if the central warehouse has a long lead time (which limits the speed to react on an imbalance situation), and if there is one local warehouse with a low mean demand and a very low additional holding cost parameter and another local warehouse with a higher mean demand and a much larger additional holding cost parameter. These results extend earlier results by Zipkin [70].

In a subsequent study, Doğru [21] (Chapter 4) computed the optimal policy via stochastic dynamic programming for a two-echelon distribution system with discrete demand distributions on small, finite supports. He compared the decisions taken under the optimal policy to the decisions taken under the LB heuristic for instances with large $(UB - C_P)/C_P$ ratios. This showed that in these instances, the allocation functions $z_n^*(\cdot)$ as used by the LB heuristic are fine, but that the aggregate base-stock level S_{loc} is too high or the S_0 is somewhat too low (both lead to a too-low average physical stock in the central warehouse). This suggests that in instances for which the LB heuristic performs poorly, a much better heuristic may be obtained by slightly adapting the base-stock levels S_0 and S_{loc}. One may even go further, and enumerate over all possible values of S_0 and S_{loc} and pick the combination with the lowest costs. That results in the DS heuristic as proposed by Gallego et al. [36], in a continuous review setting with Poisson demand processes. For this DS heuristic, small gaps between the average costs of the DS heuristic and the lower-bound LB were found. The experiments in both Doğru [21] and Gallego et al. [36] show that it makes sense to use base-stock policies in combination with the optimal allocation functions $z_n^*(\cdot)$. However, in several cases, we cannot use the levels of the LB heuristic, and we have to try other combinations. The latter increases the computational complexity, especially for systems with multiple echelon levels and many stockpoints.

Another way to cope with possible imbalance problems is by the assumption of alternative allocation rules. One such rule is FCFS allocation in distribution systems with continuous review, as used, for example, by Axsäter [2] and Sherbrooke [56]. In addition, one assumes base-stock policies. Then, the problem is to evaluate the system under a given base-stock policy and to optimize the base-stock levels. There has been much research in this direction; for an overview, see Axsäter [4]. Gallego et al. [36] executed an experiment in which a system with optimal allocation has been compared to a system with FCFS allocation. Optimal allocation always performed better, but the differences in costs were relatively small. Hence, FCFS allocation is a sensible option for systems with serious imbalance problems under the LB heuristic (distribution systems with low demand rates probably belong to this category). Other alternative allocation rules have been studied by Axsäter et al. [5] and Güllü et al. [42].

For systems without imbalance problems, the LB heuristic is appropriate. Variants of the LB heuristic have been developed to increase the speed of computational procedures. This was done by the assumption of linear instead of optimal allocation rules, and is useful for large-scale systems with multiechelon levels and many stockpoints; for research in this direction, see Diks and de Kok [20] and van der Heijden et al. [61], and the references therein.

5.3. General Systems and Connection with Practice

So far, we have treated multiechelon systems with a pure serial, a pure assembly/convergent, or a pure distribution/divergent structure. These systems are applicable in practice, for example, when a company is responsible for only a small part of the supply chain with such a pure structure and wants to control that part by multiechelon models. However, many other practical situations exist with a mixture of convergent and divergent structures. That leads to multiechelon models that are hard to solve to optimality, or to models with many stockpoints. For such models, a few interesting concepts have been developed.

There is one concept for general networks of stockpoints based on the principles for pure convergent and pure divergent systems as described in §§4.5 and 5.1. This concept is denoted as synchronized base-stock policies; for an extensive description, see de Kok and Fransoo [16]. The base-stock policies are called synchronized as the control of components that go into the same end-products are coordinated according to the insights for convergent systems. This concept has been applied at Philips Electronics to support weekly collaborative planning of operations by Philips Semiconductors and one of its customers, Philips Optical Storage; see de Kok et al. [18]. A second concept has been developed by Ettl et al. [26]. They use a continuous-review, base-stock policy for each stockpoint and assume FCFS allocation rules; this is in line with the research on continuous-review distribution systems with FCFS allocation as mentioned in §5.2. This concept has been applied at IBM; see Lin et al. [46]. A third concept for general networks has been described by Graves and Willems [40, 41] and extends earlier work by Inderfurth [43], Inderfurth and Minner [44], and Simpson [57]. This concept builds on base-stock policies, bounded demands, and decoupling of a supply chain into subsystems via safety stocks. It is mainly developed for supply chain design and has been applied at Eastman Kodak.

All three concepts have led to huge cost savings at the companies where they were applied, and, thus, these concepts have been successful already. Nevertheless, further research is desired to improve and extend them. In the first two concepts, several approximate steps are made in the evaluation of base-stock policies and optimization of base-stock levels to obtain efficient solution procedures for large networks. In the third concept, simplifying assumptions are made for the same purpose. First of all, it is relevant to study the effect of these approximations/assumptions on the quality of the generated solutions, i.e., on the distance between the generated solutions and optimal solutions (where in the case of the third concept optimal solutions for the model without simplifying assumptions are meant). Second, it would be interesting to compare these concepts for a setting in which all three concepts can be applied. Third, in all three concepts, no capacity constraints and batching rules are taken into account. If the hierarchical approach as discussed at the beginning of §1 is adopted, then one may deal with capacity issues at the first decision level via appropriately set batching rules, and at the second level decisions may be supported by multiechelon models that respect these batching rules. This suggests to incorporate insights from serial systems with fixed batch sizes and fixed replenishment intervals, cf. §§4.6 and 4.7. If the first-level decisions lead to capacity constraints (or, better workload control rules) for single or multiple items, those constraints have to be taken into account as well; although this will be hard. In fact, even single-product multiechelon models with a capacity constraint per stage are already hard (e.g., Glasserman and Tayur [37], Parker and Kapuscinski [48], and the references therein). Fourth, the first two concepts are appropriate for operational planning, but in practice they will be applied in a rolling horizon setting, and the effect of that deserves special attention.

6. A Classification of Multiechelon Systems and Conclusion

As we have seen in the previous sections, there are several multiechelon systems for which many nice results are obtained. For those systems (generalized) base-stock policies are optimal and a decomposition result applies for the optimal base-stock or reorder levels. In addition, for many of these systems, newsboy equations have been derived. Also, these systems are where newsvendor bounds (cf. Shang and Song [54, 55]) are most likely to work. We call these systems *"nice" systems*, and they are listed in the upper part of Table 1, where we distinguish two subclasses: systems for which newsboy equations have been derived and systems for which they have not been derived (at least, not yet; we believe that they do exist for these systems). The nice systems have in common that all main results are obtained via a single-cycle analysis, for which a stochastic dynamic program with a finite number of stages has to be solved. For these systems, successive cycles are more or less decoupled.

Table 1. A classification of multiechelon systems.

Nice systems

Systems for which newsboy equations have been derived:

— Standard serial system (§3.1)
— Assembly system (§4.5, Rosling [51])
— Serial system with a fixed batch size per stage (§4.6, Chen [10])
— Serial system with a fixed replenishment interval per stage (§4.7, van Houtum et al. [66])
— Distribution system under the balance assumption (§5.1)
— Serial system with a capacity constraint at the most upstream stage (§4.8, Zijm and van Houtum [69])

Systems for which no newsboy equations have been derived (at least, not yet):

— Serial system with fixed batch sizes and fixed replenishment intervals (§4.7, Chao and Zhou [8])
— Serial system with advanced demand information (§4.8, Gallego and Özer [33])
— Serial system with Markov-modulated demand (§4.8, Chen and Song [11])
— Serial system with a fixed ordering cost at the most upstream stage (§4.8, Clark and Scarf [15])

Complicated systems

— Distribution system without balance assumption (§5.2)
— Distribution systems with FCFS allocation (§5.2)
— Systems with a mixed convergent-divergent structure (§5.3)
— Systems with a capacity constraint at each stage (§5.3)
— Systems with a fixed ordering cost at each stage (§4.6, Clark and Scarf [15])

In the lower part of Table 1, we have listed a number of systems that we call *"complicated" systems*. For these systems, there is a kind of coupling (or, dependence) between successive cycles. The structure of optimal policies cannot be derived via a single-cycle analysis. Also, that structure may be rather complicated and, thus, unattractive for practical purposes. For these systems, it may be sensible (and justified) to assume (generalized) base-stock policies, as in the concepts for general systems that we discussed in §5.3. But there is no decomposition result anymore, and, thus, optimal base-stock levels have to be determined in an alternative way. In fact, even an evaluation of a base-stock policy may already be complicated.

The distinction between nice and complicated systems is delicate (as delicate as between product-form and nonproduct-form networks in the area of queueing networks). Apart from the issues raised at the end of §5.3, future research may be devoted to that distinction as well. That may lead to a bigger set of nice systems and improved insights for heuristic solutions for complicated systems.

References

[1] I. J. B. F. Adan, M. J. A. van Eenige, and J. A. C. Resing. Fitting discrete distributions on the first two moments. *Probability in the Engineering and Informational Sciences* 9:623–632, 1996.

[2] S. Axsäter. Simple solution procedures for a class of two-echelon inventory problems. *Operations Research* 38:64–69, 1990.

[3] S. Axsäter. *Inventory Control.* Kluwer, Boston, MA, 2000.

[4] S. Axsäter. Supply chain operations: Serial and distribution inventory systems, Ch. 10. A. G. de Kok and S. C. Graves, eds. *Supply Chain Management: Design, Coordination and Operation.* Handbooks in OR & MS. Elsevier, Amsterdam, The Netherlands, 2003.

[5] S. Axsäter, J. Marklund, and E. A. Silver. Heuristic methods for centralized control of one-warehouse, N-retailer inventory systems. *Manufacturing & Service Operations Management* 4:75–97, 2002.

[6] T. Boyaci and G. Gallego. Serial production/distribution systems under service constraints. *Manufacturing & Service Operations Management* 3:43–50, 2001.

[7] T. Boyaci, G. Gallego, K. H. Shang, and J. S. Song. Erratum to bounds in Serial production/ distribution systems under service constraints. *Manufacturing & Service Operations Management* 5:372–374, 2003.

[8] X. Chao and S. X. Zhou. Optimal policies for multi-echelon inventory system with batch ordering and periodic batching. Working paper, North Carolina State University, Raleigh, NC, 2005.

[9] F. Chen. Echelon reorder points, installation reorder points, and the value of centralized demand information. *Management Science* 44:S221–S234, 1998.

[10] F. Chen. Optimal policies for multi-echelon inventory problems with batch ordering. *Operations Research* 48:376–389, 2000.

[11] F. Chen and J. S. Song. Optimal policies for multiechelon inventory problems with Markov-modulated demand. *Operations Research* 49:226–234, 2001.

[12] F. Chen and Y. S. Zheng. Lower bounds for multi-echelon stochastic inventory problems. *Management Science* 40:1426–1443, 1994.

[13] A. J. Clark. A dynamic, single-item, multi-echelon inventory model. Research report, RAND Corporation, Santa Monica, CA, 1958.

[14] A. J. Clark and H. Scarf. Optimal policies for a multi-echelon inventory problem. *Management Science* 6:475–490, 1960.

[15] A. J. Clark and H. Scarf. Approximate solutions to a simple multi-echelon inventory problem, K. J. Arrow, S. Karlin, and H. Scarf, eds. *Studies in Applied Probability and Management Science*. Stanford University Press, Stanford, CA, 88–100, 1962.

[16] A. G. de Kok and J. C. Fransoo. Planning supply chain operations: Definition and comparison of planning concepts, Ch. 12. A. G. de Kok and S. C. Graves, eds. *Supply Chain Management: Design, Coordination and Cooperation*. Handbooks in OR & MS. Elsevier, Amsterdam, The Netherlands, 2003.

[17] A. G. de Kok and S. C. Graves, eds. *Supply Chain Management: Design, Coordination and Cooperation*. Handbooks in OR & MS. Elsevier, Amsterdam, The Netherlands, 2003.

[18] A. G. de Kok, F. Janssen, J. van Doremalen, E. van Wachem, M. Clerkx, and W. Peeters. Philips Electronics synchronizes its supply chain to end the bullwhip effect. *Interfaces* 35:37–48, 2005.

[19] E. B. Diks and A. G. de Kok. Optimal control of a divergent multi-echelon inventory system. *European Journal of Operational Research* 111:75–97, 1998.

[20] E. B. Diks and A. G. de Kok. Computational results for the control of a divergent N-echelon inventory system. *International Journal of Production Economics* 59:327–336, 1999.

[21] M. K. Doğru. Optimal control of one-warehouse multi-retailer systems: An assessment of the balance assumption. Ph.D. thesis, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, 2006.

[22] M. K. Doğru, A. G. de Kok, and G. J. van Houtum. A numerical study on the effect of the balance assumption in one-warehouse multi-retailer inventory systems. Working paper, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, 2006.

[23] M. K. Doğru, A. G. de Kok, and G. J. van Houtum. Newsvendor characterizations for one-warehouse multi-retailer systems with discrete demand. Working paper, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, 2006.

[24] M. K. Doğru, G. J. van Houtum, and A. G. de Kok. Newsboy equations for optimal reorder levels of serial inventory systems with fixed batch sizes. Working paper, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, 2006.

[25] G. Eppen and L. Schrage. Centralized ordering policies in a multi-warehouse system with lead times and random demand. L. B. Schwartz, ed., *Multi-Level Production/Inventory Control Systems: Theory and Practice*. North-Holland, Amsterdam, The Netherlands, 51–67, 1981.

[26] M. Ettl, G. E. Feigin, G. Y. Lin, and D. D. Yao. A supply network model with base-stock control and service requirements. *Operations Research* 48:216–232, 2000.

[27] H. Everett, III. Generalized Lagrange multiplier method for solving problems of optimum allocation of resources. *Operations Research* 11:399–417, 1963.

[28] A. Federgruen and P. H. Zipkin. Allocation policies and cost approximations for multilocation inventory systems. *Management Science* 30:69–84, 1984.

[29] A. Federgruen and P. H. Zipkin. Computational issues in an infinite horizon, multi-echelon inventory model. *Operations Research* 32:818–836, 1984.

[30] A. Federgruen and P. H. Zipkin. An inventory model with limited production capacity and uncertain demands, I. The average cost criterion. *Mathematics of Operations Research* 11:193–207, 1986.

[31] A. Federgruen and P. H. Zipkin. An inventory model with limited production capacity and uncertain demands, II. The discounted cost criterion. *Mathematics of Operations Research* 11:208–216, 1986.

[32] K. Feng and U. S. Rao. Echelon-stock (R, nT) control in two-stage serial stochastic inventory systems. *Operations Research Letters*. Forthcoming. 2006.

[33] G. Gallego and Ö. Özer. Optimal replenishment policies for multiechelon inventory problems under advance demand information. *Manufacturing & Service Operations Management* 5:157–175, 2003.

[34] G. Gallego and Ö. Özer. A new algorithm and a new heuristic for serial supply systems. *Operations Research Letters* 33:349–362, 2005.

[35] G. Gallego and P. H. Zipkin. Stock positioning and performance estimation in serial production-transportation systems. *Manufacturing & Service Operations Management* 1:77–88, 1999.

[36] G. Gallego, Ö. Özer, and P. H. Zipkin. Bounds, heuristics, and approximations for distribution systems. *Operations Research*. Forthcoming. 2006.

[37] P. Glasserman and S. R. Tayur. Sensitivity analysis for base-stock levels in multiechelon production-inventory systems. *Management Science* 41:263–281, 1995.

[38] L. Gong, A. G. de Kok, and J. Ding. Optimal leadtimes planning in a serial production system. *Management Science* 40:629–632, 1994.

[39] S. C. Graves, A multiechelon model with fixed replenishment intervals. *Management Science* 42:1–18, 1996.

[40] S. C. Graves and S. P. Willems. Optimizing strategic safety stock placement in supply chains. *Manufacturing & Service Operations Management* 2:68–83, 2000.

[41] S. C. Graves and S. P. Willems. Erratum: Optimizing strategic safety stock placement in supply chains. *Manufacturing & Service Operations Management* 5:176–177, 2003.

[42] R. Güllü, G. J. van Houtum, F. Z. Sargut, and N. K. Erkip. Analysis of a decentralized supply chain under partial cooperation. *Manufacturing & Service Operations Management* 7:229–247, 2005.

[43] K. Inderfurth. Safety stock optimization in multi-stage inventory systems. *International Journal of Production Economics* 24:103–113, 1991.

[44] K. Inderfurth and S. Minner. Safety stocks in multi-stage inventory systems under different service levels. *European Journal of Operational Research* 106:57–73, 1998.

[45] L. J. G. Langenhoff and W. H. M. Zijm. An analytical theory of multi-echelon production/distribution systems. *Statistica Neerlandica* 44:149–174, 1990.

[46] G. Lin, M. Ettl, S. Buckley, S. Bagchi, D. D. Yao, B. L. Naccarato, R. Allan, K. Kim, and L. Koenig. Extended-enterprise supply-chain management at IBM Personal Systems Group and other divisions. *Interfaces* 30:7–25, 2000.

[47] T. Osogami and M. Harchol-Balter. Closed form solutions for mapping general distributions to quasi-minimal PH distributions. *Performance Evaluation* 63:524–552, 2006.

[48] R. P. Parker and R. Kapuscinski. Optimal policies for a capacitated two-echelon inventory system. *Operations Research* 52:739–755, 2004.

[49] E. L. Porteus. *Foundations of Stochastic Inventory Theory*. Stanford University Press, Palo Alto, CA, 2002.

[50] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, New York, 1994.

[51] K. Rosling. Optimal inventory policies for assembly systems under random demand. *Operations Research* 37:565–579, 1989.

[52] H. Scarf. The optimality of (S, s) policies in the dynamic inventory problem, Ch. 13. K. Arrow, S. Karlin, and P. Suppes, eds. *Mathematical Methods in the Social Sciences*. Stanford University Press, Palo Alto, CA, 1960.

[53] R. Schassberger. *Warteschlangen*. Springer, Berlin, 1973.

[54] K. H. Shang and J. S. Song. Newsvendor bounds and heuristic for optimal policies in serial supply chains. *Management Science* 49:618–638, 2003.

[55] K. H. Shang and J. S. Song. Supply chains with economies of scale: Single-stage heuristic and approximations. Working paper, Duke University, Durham, NC, 2005.

[56] C. C. Sherbrooke. METRIC: A multi-echelon technique for recoverable item control. *Operations Research* 16:122–141, 1968.

[57] K. F. Simpson. In-process inventories. *Operations Research* 6:863–871, 1958.

[58] S. R. Tayur. Computing the optimal policy for capacitated inventory models. *Communications in Statistics-Stochastic Models* 9:585–598, 1993.

[59] S. R. Tayur, R. Ganeshan, and M. Magazine, eds. *Quantitative Models for Supply Chain Management.* Kluwer, Boston, MA, 1999.

[60] H. C. Tijms. *Stochastic Models: An Algorithmic Approach.* Wiley, New York, 1994.

[61] M. C. van der Heijden, E. B. Diks, and A. G. de Kok. Stock allocation in general multi-echelon distribution systems with (R, S) order-up-to policies. *International Journal of Production Economics* 49:157–174, 1997.

[62] G. J. van Houtum and W. H. M. Zijm. Computational procedures for stochastic multi-echelon production systems. *International Journal of Production Economics* 23:223–237, 1991.

[63] G. J. van Houtum and W. H. M. Zijm. Incomplete convolutions in production and inventory models. *OR Spektrum* 19:97–107, 1997.

[64] G. J. van Houtum and W. H. M. Zijm. On the relation between service and cost models for general inventory systems. *Statistica Neerlandica* 54:127–147, 2000.

[65] G. J. van Houtum, K. Inderfurth, and W. H. M. Zijm. Materials coordination in stochastic multiechelon systems. *European Journal of Operational Research* 95:1–23, 1996.

[66] G. J. van Houtum, A. Scheller-Wolf, and J. Yi. Optimal control of serial inventory systems with fixed replenishment intervals. *Operations Research.* Forthcoming. 2006.

[67] P. M. Vanden Bosch and D. C. Dietz. Scheduling and sequencing arrivals to an appointment system. *Journal of Service Research* 4:15–25, 2001.

[68] P. P. Wang. Sequencing and scheduling N customers for a stochastic server. *European Journal of Operational Research* 119:729–738, 1999.

[69] W. H. M. Zijm and G. J. van Houtum. On multi-stage production/inventory systems under stochastic demand. *International Journal of Production Economics* 35:391–400, 1994.

[70] P. H. Zipkin. On the imbalance of inventories in multi-echelon systems. *Mathematics of Operations Research* 9:402–423, 1984.

[71] P. H. Zipkin. *Foundations of Inventory Management.* Irwin/McGraw Hill, New York, 2000.

TUTORIALS IN
OPERATIONS RESEARCH
INFORMS 2006

 | ISBN 13 978-1-877640-20-9
DOI 10.1287/educ.1063.0023

Game Theory in Supply Chain Analysis*

Gérard P. Cachon and Serguei Netessine
The Wharton School, University of Pennsylvania, Philadelphia, Philadelphia 19104,
{cachon@wharton.upenn.edu, netessine@wharton.upenn.edu}

Abstract Game theory has become an essential tool in the analysis of supply chains with multiple agents, often with conflicting objectives. This chapter surveys the applications of game theory to supply chain analysis and outlines game-theoretic concepts that have potential for future application. We discuss both noncooperative and cooperative game theory in static and dynamic settings. Careful attention is given to techniques for demonstrating the existence and uniqueness of equilibrium in noncooperative games. A newsvendor game is employed throughout to demonstrate the application of various tools.

Keywords game theory; noncooperative; cooperative; equilibrium concepts

1. Introduction

Game theory (hereafter GT) is a powerful tool for analyzing situations in which the decisions of multiple agents affect each agent's payoff. As such, GT deals with interactive optimization problems. While many economists in the past few centuries have worked on what can be considered game-theoretic models, John von Neumann and Oskar Morgenstern are formally credited as the fathers of modern game theory. Their classic book "Theory of Games and Economic Behavior," (von Neumann and Morgenstern [102]), summarizes the basic concepts existing at that time. GT has since enjoyed an explosion of developments, including the concept of equilibrium by Nash [68], games with imperfect information by Kuhn [51], cooperative games by Aumann [3] and Shubik [86], and auctions by Vickrey [100] to name just a few. Citing Shubik [87], "In the '50s... game theory was looked upon as a curiosum not to be taken seriously by any behavioral scientist. By the late 1980s, game theory in the new industrial organization has taken over... game theory has proved its success in many disciplines."

This chapter has two goals. In our experience with GT problems, we have found that many of the useful theoretical tools are spread over dozens of papers and books, buried among other tools that are not as useful in supply chain management (hereafter SCM). Hence, our first goal is to construct a brief tutorial through which SCM researchers can quickly locate GT tools and apply GT concepts. Due to the need for short explanations, we omit all proofs, choosing to focus only on the intuition behind the results we discuss. Our second goal is to provide ample but by no means exhaustive references on the specific applications of various GT techniques. These references offer an in-depth understanding of an application where necessary. Finally, we intentionally do not explore the implications of GT analysis on supply chain management, but rather we emphasize the means of conducting the analysis to keep the exposition short.

* This chapter is reprinted with modifications from G. P. Cachon and S. Netessine "Game Theory in Supply Chain Analysis" in *Handbook of Quantitative Supply Chain Analysis: Modeling in the E-Business Era*, D. Simchi-Levi, S. D. Wu, and M. Shen, eds., 2004, with kind permission of Springer Science and Business Media.

1.1. Scope and Relation to the Literature

There are many GT concepts, but this chapter focuses on concepts that are particularly relevant to SCM and, perhaps, have already found their applications in the literature. We dedicate a considerable amount of space to the discussion of static noncooperative, nonzero sum games, the type of game which has received the most attention in the recent SCM literature. We also discuss cooperative games, dynamic/differential games, and games with asymmetric/incomplete information. We omit discussion of important GT concepts covered in Simchi-Levi et al. [88]: auctions in Chapters 4 and 10, principal-agent models in Chapter 3, and bargaining in Chapter 11.

The material in this chapter was collected predominantly from Friedman [37], Fudenberg and Tirole [38], Moulin [62], Myerson [66], Topkis [96], and Vives [101]. Some previous surveys of GT models in management science include Lucas's [57] survey of mathematical theory of games, Feichtinger and Jorgensen's [35] survey of differential games, and Wang and Parlar's [105] survey of static models. A recent survey by Li and Whang [55] focuses on application of GT tools in five specific OR/MS models.

2. Noncooperative Static Games

In noncooperative static games, the players choose strategies simultaneously and are thereafter committed to their chosen strategies, i.e., these are simultaneous move, one-shot games. Noncooperative GT seeks a rational prediction of how the game will be played in practice.[1] The solution concept for these games was formally introduced by John Nash [68], although some instances of using similar concepts date back a couple of centuries.

2.1. Game Setup

To break the ground for the section, we introduce basic GT notation. A warning to the reader: to achieve brevity, we intentionally sacrifice some precision in our presentation. See the texts by Friedman [37] and Fudenberg and Tirole [38] if more precision is required.

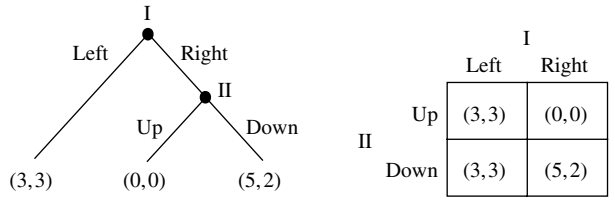
Throughout this chapter, we represent games in the normal form. A game in the normal form consists of (1) *players* indexed by $i = 1, \ldots, n$, (2) *strategies* or more generally a set of strategies denoted by x_i, $i = 1, \ldots, n$ available to each player, and (3) *payoffs* $\pi_i(x_1, x_2, \ldots, x_n)$, $i = 1, \ldots, n$ received by each player. Each strategy is defined on a set X_i, $x_i \in X_i$, so we call the Cartesian product $X_1 \times X_2 \times \cdots \times X_n$ the *strategy space*. Each player may have a unidimensional strategy or a multidimensional strategy. In most SCM applications, players have unidimensional strategies, so we shall either explicitly or implicitly assume unidimensional strategies throughout this chapter. Furthermore, with the exception of one example, we will work with continuous strategies, so the strategy space is R^n.

A player's strategy can be thought of as the complete instruction for which actions to take in a game. For example, a player can give his or her strategy to someone who has absolutely no knowledge of the player's payoff or preferences, and that person should be able to use the instructions contained in the strategy to choose the actions the player desires. As a result, each player's set of feasible strategies must be independent of the strategies chosen by the other players, i.e., the strategy choice by one player is not allowed to limit the feasible strategies of another player. (Otherwise, the game is ill defined and any analytical results obtained from the game are questionable.)

In the normal form, players choose strategies simultaneously. Actions are adopted after strategies are chosen and those actions correspond to the chosen strategies. As an alternative to the one-shot selection of strategies in the normal form, a game can also be designed in the extensive form. With the extensive form, actions are chosen only as needed, so sequential

[1]Some may argue that GT should be a tool for choosing how a manager should play a game, which may involve playing against rational or semirational players. In some sense there is no conflict between these descriptive and normative roles for GT, but this philosophical issue surely requires more in-depth treatment than can be afforded here.

FIGURE 1. Extensive vs. normal form game representation.



choices are possible. As a result, players may learn information between the selection of actions, in particular, a player may learn which actions were previously chosen or what the outcome of a random event was. Figure 1 provides an example of a simple extensive form game and its equivalent normal form representation: There are two players: player I chooses from {Left, Right} and player II chooses from {Up, Down}. In the extensive form, player I chooses first, then player II chooses after learning player I's choice. In the normal form, they choose simultaneously. The key distinction between normal and extensive form games is that in the normal form, a player is able to commit to all future decisions. We later show that this additional commitment power may influence the set of plausible equilibria.

A player can choose a particular strategy or a player can choose to randomly select from among a set of strategies. In the former case, the player is said to choose a *pure strategy*, whereas in the latter case, the player chooses a *mixed strategy*. There are situations in economics and marketing that have used mixed strategies: see Varian [99] for search models and Lal [52] for promotion models. However, mixed strategies have not been applied in SCM, in part because it is not clear how a manager would actually implement a mixed strategy. For example, it seems unreasonable to suggest that a manager should "flip a coin" among various capacity levels. Fortunately, mixed strategy equilibria do not exist in games with a unique pure strategy equilibrium. Hence, in those games, attention can be restricted to pure strategies without loss of generality. Therefore, in the remainder of this chapter, we consider only pure strategies.

In a *noncooperative* game, the players are unable to make binding commitments before choosing their strategies. In a *cooperative* game, players are able to make binding commitments. Hence, in a cooperative game, players can make side-payments and form coalitions. We begin our analysis with noncooperative static games. In all sections, except the last one, we work with the games of *complete information*, i.e., the players' strategies and payoffs are common knowledge to all players.

As a practical example throughout this chapter, we utilize the classic newsvendor problem transformed into a game. In the absence of competition, each newsvendor buys Q units of a single product at the beginning of a single selling season. Demand during the season is a random variable D with distribution function F_D and density function f_D. Each unit is purchased for c and sold on the market for $r > c$. The newsvendor solves the following optimization problem

$$\max_Q \pi = \max_Q E_D[r \min(D, Q) - cQ],$$

with the unique solution

$$Q^* = F_D^{-1}\left(\frac{r-c}{r}\right).$$

Goodwill penalty costs and salvage revenues can easily be incorporated into the analysis, but for our needs, we normalized them out.

Now consider the GT version of the newsvendor problem with two retailers competing on product availability. Parlar [75] was the first to analyze this problem, which is also one of the first articles modeling inventory management in a GT framework. It is useful to consider only the two-player version of this game because then graphic analysis and interpretations

are feasible. Denote the two players by subscripts i and j, their strategies (in this case, stocking quantities) by Q_i, Q_j, and their payoffs by π_i, π_j.

We introduce interdependence of the players' payoffs by assuming the two newsvendors sell the same product. As a result, if retailer i is out of stock, all unsatisfied customers try to buy the product at retailer j instead. Hence, retailer i's total demand is $D_i + (D_j - Q_j)^+$: the sum of his own demand and the demand from customers not satisfied by retailer j. Payoffs to the two players are then

$$\pi_i(Q_i, Q_j) = E_D[r_i \min(D_i + (D_j - Q_j)^+, Q_i) - c_i Q_i], \quad i, j = 1, 2.$$

2.2. Best Response Functions and the Equilibrium of the Game

We are ready for the first important GT concept: *Best response functions.*

Definition 1. Given an n-player game, player i's best response (function) to the strategies x_{-i} of the other players is the strategy x_i^* that maximizes player $i's$ payoff $\pi_i(x_i, x_{-i})$:

$$x_i^*(x_{-i}) = \arg\max_{x_i} \pi_i(x_i, x_{-i}).$$

($x_i^*(x_{-i})$ is probably better described as a correspondence rather than a function, but we shall nevertheless call it a function with an understanding that we are interpreting the term "function" liberally.) If π_i is quasi-concave in x_i, the best response is uniquely defined by the first-order conditions of the payoff functions. In the context of our competing newsvendors example, the best response functions can be found by optimizing each player's payoff functions w.r.t. the player's own decision variable Q_i while taking the competitor's strategy Q_j as given. The resulting best response functions are

$$Q_i^*(Q_j) = F^{-1}_{D_i + (D_j - Q_j)^+}\left(\frac{r_i - c_i}{r_i}\right), \quad i, j = 1, 2.$$

Taken together, the two best response functions form a *best response mapping* $R^2 \to R^2$, or in the more general case, $R^n \to R^n$. Clearly, the best response is the best player i can hope for given the decisions of other players. Naturally, an outcome in which all players choose their best responses is a candidate for the noncooperative solution. Such an outcome is called a Nash equilibrium (hereafter NE) of the game.

Definition 2. An outcome $(x_1^*, x_2^*, \ldots, x_n^*)$ is a Nash equilibrium of the game if x_i^* is a best response to x_{-i}^* for all $i = 1, 2, \ldots, n$.

Going back to competing newsvendors, NE is characterized by solving *a system* of best responses that translates into the system of first-order conditions:

$$Q_1^*(Q_2^*) = F^{-1}_{D_1 + (D_2 - Q_2^*)^+}\left(\frac{r_1 - c_1}{r_1}\right),$$

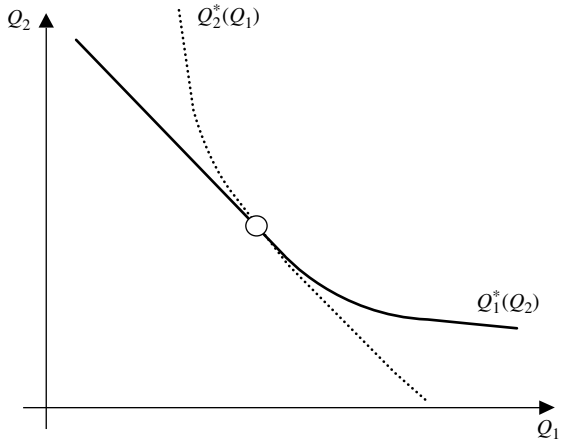
$$Q_2^*(Q_1^*) = F^{-1}_{D_2 + (D_1 - Q_1^*)^+}\left(\frac{r_2 - c_2}{r_2}\right).$$

When analyzing games with two players, it is often helpful to graph the best response functions to gain intuition. Best responses are typically defined implicitly through the first-order conditions, which makes analysis difficult. Nevertheless, we can gain intuition by finding out how each player reacts to an increase in the stocking quantity by the other player (i.e., $\partial Q_i^*(Q_j)/\partial Q_j$) through employing implicit differentiation as follows:

$$\frac{\partial Q_i^*(Q_j)}{\partial Q_j} = -\frac{\partial^2 \pi_i / \partial Q_i \partial Q_j}{\partial^2 \pi_i / \partial Q_i^2} = -\frac{r_i f_{D_i + (D_j - Q_j)^+ | D_j > Q_j}(Q_i) \Pr(D_j > Q_j)}{r_i f_{D_i + (D_j - Q_j)^+}(Q_i)} < 0. \tag{1}$$

The expression says that the *slopes* of the best response functions are negative, which implies an intuitive result that each player's best response is monotonically decreasing in the other

FIGURE 2. Best responses in the newsvendor game.



player's strategy. Figure 2 presents this result for the symmetric newsvendor game. The equilibrium is located on the intersection of the best responses, and we also see that the best responses are, indeed, decreasing.

One way to think about an NE is as a *fixed point* of the best response mapping $R^n \to R^n$. Indeed, according to the definition, NE must satisfy the system of equations $\partial\pi_i/\partial x_i = 0$, all i. Recall that a fixed point x of mapping $f(x)$, $R^n \to R^n$ is any x such that $f(x) = x$. Define $f_i(x_1, \ldots, x_n) = \partial\pi_i/\partial x_i + x_i$. By the definition of a fixed point,

$$f_i(x_1^*, \ldots, x_n^*) = x_i^* = \partial\pi_i(x_1^*, \ldots, x_n^*)/\partial x_i + x_i^* \to \partial\pi_i(x_1^*, \ldots, x_n^*)/\partial x_i = 0, \quad \forall i.$$

Hence, x^* solves the first-order conditions if and only if it is a fixed point of mapping $f(x)$ defined above.

The concept of NE is intuitively appealing. Indeed, it is a self-fulfilling prophecy. To explain, suppose a player were to guess the strategies of the other players. A guess would be consistent with payoff maximization and therefore would be reasonable only if it presumes that strategies are chosen to maximize every player's payoff given the chosen strategies. In other words, with any set of strategies that is not an NE there exists at least one player that is choosing a nonpayoff maximizing strategy. Moreover, the NE has a self-enforcing property: No player wants to unilaterally deviate from it because such behavior would lead to lower payoffs. Hence, NE seems to be the necessary condition for the prediction of any rational behavior by players.[2]

While attractive, numerous criticisms of the NE concept exist. Two particularly vexing problems are the nonexistence of equilibrium and the multiplicity of equilibria. Without the existence of an equilibrium, little can be said regarding the likely outcome of the game. If multiple equilibria exist, then it is not clear which one will be the outcome. Indeed, it is possible the outcome is not even an equilibrium because the players may choose strategies from different equilibria. For example, consider the normal form game in Figure 1. There are two Nash equilibria in that game {Left, Up} and {Right, Down}: Each is a best response to the other player's strategy. However, because the players choose their strategies simultaneously, it is possible that player I chooses Right (the second equilibrium) while player II chooses Up (the first equilibrium), which results in {Right, Up}, the worst outcome for both players.

[2]However, an argument can also be made that to predict rational behavior by players it is sufficient that players not choose dominated strategies, where a dominated strategy is one that yields a lower payoff than some other strategy (or convex combination of other strategies) for all possible strategy choices by the other players.

In some situations, it is possible to rationalize away some equilibria via a refinement of the NE concept: e.g., trembling hand perfect equilibrium (Selten [83]), sequential equilibrium (Kreps and Wilson [50]), and proper equilibria (Myerson [66]). These refinements eliminate equilibria that are based on noncredible threats, i.e., threats of future actions that would not actually be adopted if the sequence of events in the game led to a point in the game in which those actions could be taken. The extensive form game in Figure 1 illustrates this point. {Left, Up} is a Nash equilibrium (just as it is in the comparable normal form game) because each player is choosing a best response to the other player's strategy: Left is optimal for player I given player II plans to play Up and player II is indifferent between Up or Down given player I chooses Left. But if player I were to choose Right, then it is unreasonable to assume player II would actually follow through with UP: UP yields a payoff of 0 while Down yields a payoff of 2. Hence, the {Left, Up} equilibrium is supported by a noncredible threat by player II to play Up. Although these refinements are viewed as extremely important in economics (Selten was awarded the Nobel Prize for his work), the need for these refinements has not yet materialized in the SCM literature. However, that may change as more work is done on sequential/dynamic games.

An interesting feature of the NE concept is that the system optimal solution (i.e., a solution that maximizes the sum of players' payoffs) need not be an NE. Hence, decentralized decision making generally introduces inefficiency in the supply chain. There are, however, some exceptions: see Mahajan and van Ryzin [59] and Netessine and Zhang [73] for situations in which competition may result in the system-optimal performance. In fact, an NE may not even be on the *Pareto frontier*: The set of strategies such that each player can be made better off only if some other player is made worse off. A set of strategies is *Pareto optimal* if they are on the Pareto frontier; otherwise, a set of strategies is *Pareto inferior*. Hence, an NE can be Pareto inferior. The prisoner's dilemma game (Fudenberg and Tirole [38]) is the classic example of this: Only one pair of strategies when both players "cooperate" is Pareto optimal, and the unique Nash equilibrium is when both players "defect" happens to be Pareto inferior. A large body of the SCM literature deals with ways to align the incentives of competitors to achieve optimality. See Cachon [17] for a comprehensive survey and taxonomy. See Cachon [18] for a supply chain analysis that makes extensive use of the Pareto optimal concept.

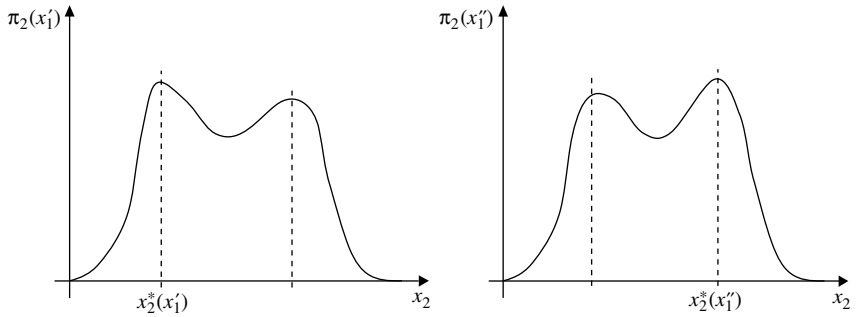
2.3. Existence of Equilibrium

An NE is a solution to a system of n first-order conditions; therefore, an equilibrium may not exist. Nonexistence of an equilibrium is potentially a conceptual problem because in this case the outcome of the game is unclear. However, in many games an NE does exist, and there are some reasonably simple ways to show that at least one NE exists. As already mentioned, an NE is a fixed point of the best response mapping. Hence, fixed-point theorems can be used to establish the existence of an equilibrium. There are three key fixed-point theorems, named after their creators: Brouwer, Kakutani, and Tarski, see Border [13] for details and references. However, direct application of fixed-point theorems is somewhat inconvenient, and hence generally not done. For exceptions, see Lederer and Li [54] and Majumder and Groenevelt [60] for existence proofs that are based on Brouwer's fixed point theorem. Alternative methods, derived from these fixed-point theorems, have been developed. The simplest and the most widely used technique for demonstrating the existence of NE is through verifying concavity of the players' payoffs.

Theorem 1 (Debreu [29]). *Suppose that for each player, the strategy space is compact*[3] *and convex and the payoff function is continuous and quasiconcave with respect to each player's own strategy. Then, there exists at least one pure strategy NE in the game.*

[3]Strategy space is compact if it is closed and bounded.

FIGURE 3. Example with a bimodal objective function.



If the game is symmetric in a sense that the players' strategies and payoffs are identical, one would imagine that a symmetric solution should exist. This is indeed the case, as the next Theorem ascertains.

Theorem 2. *Suppose that a game is symmetric, and for each player, the strategy space is compact and convex and the payoff function is continuous and quasiconcave with respect to each player's own strategy. Then, there exists at least one symmetric pure strategy NE in the game.*

To gain some intuition about why nonquasiconcave payoffs may lead to nonexistence of NE, suppose that in a two-player game, player 2 has a bimodal objective function with two local maxima. Furthermore, suppose that a small change in the strategy of player 1 leads to a shift of the global maximum for player 2 from one local maximum to another. To be more specific, let us say that at x_1', the global maximum $x_2^*(x_1')$ is on the left (Figure 3 left) and at x_1'', the global maximum $x_2^*(x_2'')$ is on the right (Figure 3 right). Hence, a small change in x_1 from x_1' to x_1'' induces a jump in the best response of player 2, x_2^*. The resulting best response mapping is presented in Figure 4, and there is no NE in pure strategies in this game. In other words, best response functions do not intersect anywhere. As a more specific example, see Netessine and Shumsky [72] for an extension of the newsvendor game to the situation in which product inventory is sold at two different prices; such a game may not have an NE because both players' objectives may be bimodal. Furthermore, Cachon and Harker [20] demonstrate that pure strategy NE may not exist in two other important settings: Two retailers competing with cost functions described by the economic order quantity (EOQ) model, or two service providers competing with service times described by the $M/M/1$ queuing model.

The assumption of a compact strategy space may seem restrictive. For example, in the newsvendor game, the strategy space R_+^2 is not bounded from above. However, we could

FIGURE 4. Nonexistence of NE.

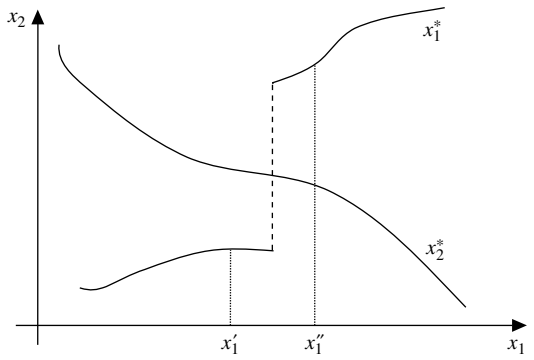
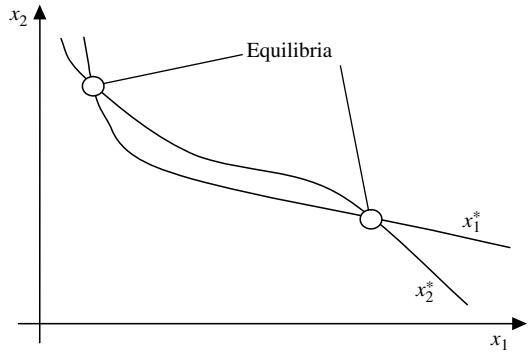


FIGURE 5. Nonuniqueness of the equilibrium.



easily bound it with some large enough finite number to represent the upper bound on the demand distribution. That bound would not impact any of the choices, and, therefore, the transformed game behaves just as the original game with an unbounded strategy space. (However, that bound cannot depend on any player's strategy choice.)

To continue with the newsvendor game analysis, it is easy to verify that the newsvendor's objective function is concave and, hence, quasiconcave w.r.t. the stocking quantity by taking the second derivative. Hence, the conditions of Theorem 1 are satisfied, and an NE exists. There are virtually dozens of papers employing Theorem 1. See, for example, Lippman and McCardle [56] for the proof involving quasiconcavity, and Mahajan and van Ryzin [58] and Netessine et al. [74] for the proofs involving concavity. Clearly, quasiconcavity of each player's objective function only implies uniqueness of the best response but does not imply a unique NE. One can easily envision a situation in which unique best response functions cross more than once so that there are multiple equilibria (see Figure 5).

If quasiconcavity of the players' payoffs cannot be verified, there is an alternative existence proof that relies on Tarski's [93] fixed-point theorem and involves the notion of supermodular games. The theory of supermodular games is a relatively recent development introduced and advanced by Topkis [96].

Definition 3. A twice continuously differentiable payoff function $\pi_i(x_1, \ldots, x_n)$ is supermodular (submodular) iff $\partial^2\pi_i/\partial x_i \partial x_j \geq 0$ (≤ 0) for all x and all $j \neq i$. The game is called supermodular if the players' payoffs are supermodular.

Supermodularity essentially means complementarity between any two strategies and is not linked directly to either convexity, concavity, or even continuity. (This is a significant advantage when forced to work with discrete strategies, e.g., Cachon [16].) However, similar to concavity/convexity, supermodularity/submodularity is preserved under maximization, limits, and addition and, hence, under expectation/integration signs, an important feature in stochastic SCM models. While in most situations the positive sign of the second derivative can be used to verify supermodularity (using Definition 3), sometimes it is necessary to utilize supermodularity-preserving transformations to show that payoffs are supermodular. Topkis [96] provides a variety of ways to verify that the function is supermodular, and some of these results are used in Cachon and Lariviere [22], Corbett [26], Netessine and Rudi [69, 71]. The following theorem follows directly from Tarski's fixed-point result and provides another tool to show existence of NE in noncooperative games:

Theorem 3. *In a supermodular game, there exists at least one NE.*

Coming back to the competitive newsvendors example, recall that the second-order cross-partial derivative was found to be

$$\frac{\partial^2\pi_i}{\partial Q_i \partial Q_j} = -r_i f_{D_i+(D_j-Q_j)^+|D_j>Q_j}(Q_i)\Pr(D_j > Q_j) < 0,$$

so that the newsvendor game is submodular, and, hence, existence of equilibrium cannot be assured. However, a standard trick is to redefine the ordering of the players' strategies. Let $y = -Q_j$ so that

$$\frac{\partial^2 \pi_i}{\partial Q_i \partial y} = r_i f_{D_i+(D_j+y)^+|D_j>Q_j}(Q_i)\Pr(D_j > -y) > 0,$$

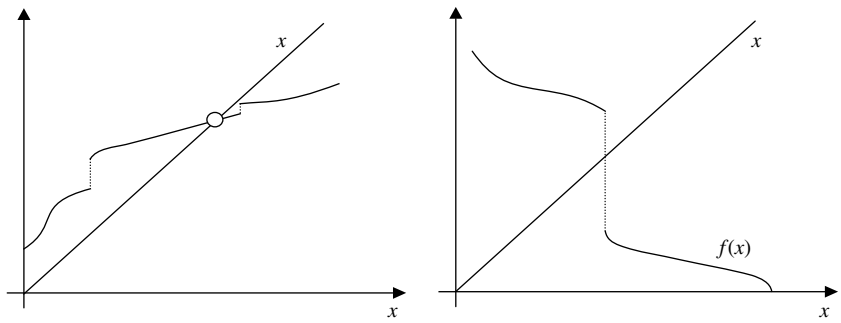
and the game becomes supermodular in (x_i, y), therefore, existence of NE is assured. Notice that we do not change either payoffs or the structure of the game, we only alter the ordering of one player's strategy space. Obviously, this trick only works in two-player games, see also Lippman and McCardle [56] for analysis of the more general version of the newsvendor game using a similar transformation. Hence, we can state that, in general, NE exists in games with decreasing best responses (submodular games) with two players. This argument can be generalized slightly in two ways that we mention briefly, see Vives [101] for details. One way is to consider an n-player game in which best responses are functions of aggregate actions of all other players, that is, $x_i^* = x_i^*(\sum_{j\neq i} x_j)$. If best responses in such a game are decreasing, then NE exists. Another generalization is to consider the same game with $x_i^* = x_i^*(\sum_{j\neq i} x_j)$ but require symmetry. In such a game, existence can be shown even with nonmonotone best responses, provided that there are only jumps up, but on intervals between jumps, best responses can be increasing or decreasing.

We now step back to discuss the intuition behind the supermodularity results. Roughly speaking, Tarski's fixed-point theorem only requires best response mappings to be nondecreasing for the existence of equilibrium and does not require quasiconcavity of the players' payoffs and allows for jumps in best responses. While it may be hard to believe that nondecreasing best responses is the only requirement for the existence of an NE, consider once again the simplest form of a single-dimensional equilibrium as a solution to the fixed-point mapping $x = f(x)$ on the compact set. It is easy to verify after a few attempts that if $f(x)$ is nondecreasing but possibly with jumps up, then it is not possible to derive a situation without an equilibrium. However, when $f(x)$ jumps down, nonexistence is possible (see Figure 6).

Hence, increasing best response functions is the only major requirement for an equilibrium to exist; players' objectives do not have to be quasiconcave or even continuous. However, to describe an existence theorem with noncontinuous payoffs requires the introduction of terms and definitions from lattice theory. As a result, we restricted ourselves to the assumption of continuous payoff functions, and in particular, to twice-differentiable payoff functions.

Although it is now clear why increasing best responses ensure existence of an equilibrium, it is not immediately obvious why Definition 3 provides a sufficient condition, given that it only concerns the sign of the second-order cross-partial derivative. To see this connection, consider separately the continuous and the discontinuous parts of the best response $x_i^*(x_j)$.

FIGURE 6. Increasing (left) and decreasing (right) mappings.



When the best response is continuous, we can apply the implicit function theorem to find its slope as follows

$$\frac{\partial x_i^*}{\partial x_j} = -\frac{\partial^2 \pi_i / \partial x_i \partial x_j}{\partial^2 \pi_i / \partial x_i^2}.$$

Clearly, if x_i^* is the best response, it must be the case that $\partial^2 \pi_i / \partial x_i^2 < 0$ or else it would not be the best response. Hence, for the slope to be positive, it is sufficient to have $\partial^2 \pi_i / \partial x_i \partial x_j > 0$, which is what Definition 3 provides. This reasoning does not, however, work at discontinuities in best responses because the implicit function theorem cannot be applied. To show that only jumps up are possible if $\partial^2 \pi_i / \partial x_i \partial x_j > 0$ holds, consider a situation in which there is a jump down in the best response. As one can recall, jumps in best responses happen when the objective function is bimodal (or more generally multimodal). For example, consider a specific point $x_j^{\#}$ and let $x_i^1(x_j^{\#}) < x_i^2(x_j^{\#})$ be two distinct points at which first-order conditions hold (i.e., the objective function π_i is bimodal). Further, suppose $\pi_i(x_i^1(x_j^{\#}), x_j^{\#}) < \pi_i(x_i^2(x_j^{\#}), x_j^{\#})$, but $\pi_i(x_i^1(x_j^{\#} + \varepsilon), x_j^{\#} + \varepsilon) > \pi_i(x_i^2(x_j^{\#} + \varepsilon), x_j^{\#} + \varepsilon)$. That is, initially, $x_i^2(x_j^{\#})$ is a global maximum, but as we increase $x_j^{\#}$ infinitesimally, there is a *jump down*, and a smaller $x_i^1(x_j^{\#} + \varepsilon)$ becomes the global maximum. For this to be the case, it must be that

$$\frac{\partial \pi_i(x_i^1(x_j^{\#}), x_j^{\#})}{\partial x_j} > \frac{\partial \pi_i(x_i^2(x_j^{\#}), x_j^{\#})}{\partial x_j},$$

or, in words, the objective function rises faster at $(x_i^1(x_j^{\#}), x_j^{\#})$ than at $(x_i^2(x_j^{\#}), x_j^{\#})$. This, however, can only happen if $\partial^2 \pi_i / \partial x_i \partial x_j < 0$ at least somewhere on the interval $[x_i^1(x_j^{\#}), x_i^2(x_j^{\#})]$, which is a contradiction. Hence, if $\partial^2 \pi_i / \partial x_i \partial x_j > 0$ holds, then only jumps up in the best response are possible.

2.4. Uniqueness of Equilibrium

From the perspective of generating qualitative insights, it is quite useful to have a game with a unique NE. If there is only one equilibrium, then one can characterize equilibrium actions without much ambiguity. Unfortunately, demonstrating uniqueness is generally much harder than demonstrating existence of equilibrium. This section provides several methods for proving uniqueness. No single method dominates; all may have to be tried to find the one that works. Furthermore, one should be careful to recognize that these methods assume existence, i.e., existence of NE must be shown separately. Finally, it is worth pointing out that uniqueness results are only available for games with continuous best response functions and, hence, there are no general methods to prove uniqueness of NE in supermodular games.

2.4.1. Method 1. Algebraic Argument. In some rather fortunate situations, one can ascertain that the solution is unique by simply looking at the optimality conditions. For example, in a two-player game, the optimality condition of one player may have a unique closed-form solution that does not depend on the other player's strategy, and, given the solution for one player, the optimality condition for the second player can be solved uniquely (Hall and Porteus [43], Netessine and Rudi [70]). In other cases, one can assure uniqueness by analyzing geometrical properties of the best response functions and arguing that they intersect only once. Of course, this is only feasible in two-player games. See Parlar [75] for a proof of uniqueness in the two-player newsvendor game and Majumder and Groenevelt [61] for a supply chain game with competition in reverse logistics. However, in most situations, these geometrical properties are also implied by the more formal arguments stated below. Finally, it may be possible to use a contradiction argument: Assume that there is more than one equilibrium and prove that such an assumption leads to a contradiction, as in Lederer and Li [54].

2.4.2. Method 2. Contraction Mapping Argument. Although the most restrictive among all methods, the contraction mapping argument is the most widely known and is the most frequently used in the literature because it is the easiest to verify. The argument is based on showing that the best response mapping is a contraction, which then implies the mapping has a unique fixed point. To illustrate the concept of a contraction mapping, suppose we would like to find a solution to the following fixed point equation:

$$x = f(x), \quad x \in R^1.$$

To do so, a sequence of values is generated by an iterative algorithm, $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots\}$ where $x^{(1)}$ is arbitrarily picked and $x^{(t)} = f(x^{(t-1)})$. The hope is that this sequence converges to a unique fixed point. It does so if, roughly speaking, each step in the sequence moves closer to the fixed point. One could verify that if $|f'(x)| < 1$ in some vicinity of x^*, then such an iterative algorithm converges to a unique $x^* = f(x^*)$. Otherwise, the algorithm diverges. Graphically, the equilibrium point is located on the intersection of two functions: x and $f(x)$. The iterative algorithm is presented in Figure 7. The iterative scheme in Figure 7 left is a contraction mapping: It approaches the equilibrium after every iteration.

Definition 4. Mapping $f(x)$, $R^n \to R^n$ is a contraction iff $\|f(x_1) - f(x_2)\| \leq \alpha \|x_1 - x_2\|$, $\forall x_1, x_2, \alpha < 1$.

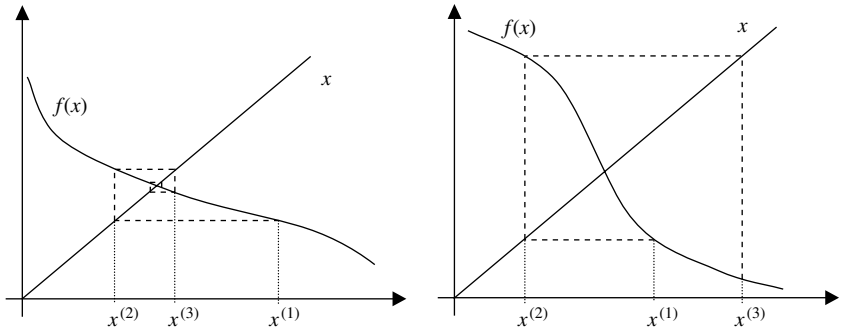
In words, the application of a contraction mapping to any two points strictly reduces (i.e., $\alpha = 1$ does not work) the distance between these points. The norm in the definition can be any norm, i.e., the mapping can be a contraction in one norm and not a contraction in another norm.

Theorem 4. *If the best response mapping is a contraction on the entire strategy space, there is a unique NE in the game.*

One can think of a contraction mapping in terms of iterative play: Player 1 selects some strategy, then player 2 selects a strategy based on the decision by player 1, etc. If the best response mapping is a contraction, the NE obtained as a result of such iterative play is *stable* but the opposite is not necessarily true; i.e., no matter where the game starts, the final outcome is the same. See also Moulin [62] for an extensive treatment of stable equilibria.

A major restriction in Theorem 4 is that the contraction mapping condition must be satisfied everywhere. This assumption is quite restrictive because the best response mapping may be a contraction locally, say, in some not necessarily small ε-neighborhood of the equilibrium, but not outside of it. Hence, if iterative play starts in this ε-neighborhood, then it converges to the equilibrium, but starting outside that neighborhood may not lead to the equilibrium (even if the equilibrium is unique). Even though one may wish to argue that it is reasonable for the players to start iterative play close to the equilibrium, formalization of such an argument is rather difficult. Hence, we must impose the condition that the entire

FIGURE 7. Converging (left) and diverging (right) iterations.



strategy space be considered. See Stidham [90] for an interesting discussion of stability issues in a queuing system.

While Theorem 4 is a starting point toward a method for demonstrating uniqueness, it does not actually explain how to validate that a best reply mapping is a contraction. Suppose we have a game with n players each endowed with the strategy x_i and we have obtained the best response functions for all players, $x_i = f_i(x_{-i})$. We can then define the following matrix of derivatives of the best response functions:

$$A = \begin{vmatrix} 0 & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & 0 & \cdots & \frac{\partial f_2}{\partial x_2} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & 0 \end{vmatrix}.$$

Further, denote by $\rho(A)$ the spectral radius of matrix A and recall that the spectral radius of a matrix is equal to the largest absolute eigenvalue $\rho(A) = \{\max |\lambda| : Ax = \lambda x, x \neq 0\}$ (Horn and Johnson [46]).

Theorem 5. *The mapping $f(x)$: $R^n \to R^n$ is a contraction if and only if $\rho(A) < 1$ everywhere.*

Theorem 5 is simply an extension of the iterative convergence argument we used above into multiple dimensions, and the spectral radius rule is an extension of the requirement $|f'(x)| < 1$. Still, Theorem 5 is not as useful as we would like it to be: Calculating eigenvalues of a matrix is not trivial. Instead, it is often helpful to use the fact that the largest eigenvalue and, hence, the spectral radius is bounded above by any of the matrix norms (Horn and Johnson [46]). So, instead of working with the spectral radius itself, it is sufficient to show $\|A\| < 1$ for any one matrix norm. The most convenient matrix norms are the maximum column-sum and the maximum row-sum norms (see Horn and Johnson [46] for other matrix norms). To use either of these norms to verify the contraction mapping, it is sufficient to verify that no column sum or no row sum of matrix A exceeds 1,

$$\sum_{i=1}^{n} \left| \frac{\partial f_k}{\partial x_i} \right| < 1 \quad \text{or} \quad \sum_{i=1}^{n} \left| \frac{\partial f_i}{\partial x_k} \right| < 1, \qquad \forall k.$$

Netessine and Rudi [69] used the contraction mapping argument in this most general form in the multiple-player variant of the newsvendor game described above.

A challenge associated with the contraction mapping argument is finding best response functions, because in most SC models, best responses cannot be found explicitly. Fortunately, Theorem 5 only requires the derivatives of the best response functions, which can be done using the implicit function theorem (from now on, IFT, see Bertsekas [12]). Using the IFT, Theorem 5 can be restated as

$$\sum_{i=1,\, i\neq k}^{n} \left| \frac{\partial^2 \pi_k}{\partial x_k \partial x_i} \right| < \left| \frac{\partial^2 \pi_k}{\partial x_k^2} \right|, \quad \forall k. \tag{2}$$

This condition is also known as "diagonal dominance" because the diagonal of the matrix of second derivatives, also called the Hessian, dominates the off-diagonal entries:

$$H = \begin{vmatrix} \frac{\partial^2 \pi_1}{\partial x_1^2} & \frac{\partial^2 \pi_1}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 \pi_1}{\partial x_1 \partial x_n} \\ \frac{\partial^2 \pi_2}{\partial x_2 \partial x_1} & \frac{\partial^2 \pi_2}{\partial x_2^2} & \cdots & \frac{\partial^2 \pi_1}{\partial x_2 \partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 \pi_n}{\partial x_n \partial x_1} & \frac{\partial^2 \pi_n}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 \pi_n}{\partial x_n^2} \end{vmatrix}. \tag{3}$$

Contraction mapping conditions in the diagonal dominance form have been used extensively by Bernstein and Federgruen [7, 8, 9, 11]. As has been noted by Bernstein and Federgruen [10], many standard economic demand models satisfy this condition.

In games with only two players, the condition in Theorem 5 simplifies to

$$\left|\frac{\partial f_1}{\partial x_2}\right| < 1 \quad \text{and} \quad \left|\frac{\partial f_2}{\partial x_1}\right| < 1, \tag{4}$$

i.e., the slopes of the best response functions are less than one. This condition is especially intuitive if we use the graphic illustration (Figure 2). Given that the slope of each best response function is less than one everywhere, if they cross at one point then they cannot cross at an additional point. A contraction mapping argument in this form was used by Van Mieghem [97] and by Rudi et al. [81].

Returning to the newsvendor game example, we have found that the slopes of the best response functions are

$$\left|\frac{\partial Q_i^*(Q_j)}{\partial Q_j}\right| = \left|\frac{f_{D_i+(D_j-Q_j)^+|D_j>Q_j}(Q_i)\Pr(D_j > Q_j)}{f_{D_i+(D_j-Q_j)^+}(Q_i)}\right| < 1.$$

Hence, the best response mapping in the newsvendor game is a contraction, and the game has a unique and stable NE.

2.4.3. Method 3. Univalent Mapping Argument. Another method for demonstrating uniqueness of equilibrium is based on verifying that the best response mapping is one to one: That is, if $f(x)$ is a $R^n \to R^n$ mapping, then $y = f(x)$ implies that for all $x' \neq x$, $y \neq f(x')$. Clearly, if the best response mapping is one to one, then there can be at most one fixed point of such mapping. To make an analogy, recall that, if the equilibrium is interior,[4] the NE is a solution to the system of the first-order conditions: $\partial \pi_i / \partial x_i = 0$, $\forall i$, which defines the best response mapping. If this mapping is single-dimensional $R^1 \to R^1$, then it is quite clear that the condition sufficient for the mapping to be one to one is quasiconcavity of π_i. Similarly, for the $R^n \to R^n$ mapping to be one to one, we require quasiconcavity of the mapping, which translates into quasidefiniteness of the Hessian:

Theorem 6. *Suppose the strategy space of the game is convex and all equilibria are interior. Then, if the determinant $|H|$ is negative quasidefinite (i.e., if the matrix $H + H^T$ is negative definite) on the players' strategy set, there is a unique NE.*

[4]Interior equilibrium is the one in which first-order conditions hold for each player. The alternative is boundary equilibrium in which at least one of the players select the strategy on the boundary of his strategy space.

Proof of this result can be found in Gale and Nikaido [40] and some further developments that deal with boundary equilibria are found in Rosen [80]. Notice that the univalent mapping argument is somewhat weaker than the contraction mapping argument. Indeed, the restatement (2) of the contraction mapping theorem directly implies univalence because the dominant diagonal assures us that H is negative definite. Hence, it is negative quasidefinite. It immediately follows that the newsvendor game satisfies the univalence theorem. However, if some other matrix norm is used, the relationship between the two theorems is not that specific. In the case of just two players, the univalence theorem can be written as, according to Moulin [62],

$$\left|\frac{\partial^2\pi_2}{\partial x_2\partial x_1}+\frac{\partial^2\pi_1}{\partial x_1\partial x_2}\right|\le 2\sqrt{\left|\frac{\partial^2\pi_1}{\partial x_1^2}\cdot\frac{\partial^2\pi_2}{\partial x_2^2}\right|},\quad \forall x_1,x_2.$$

2.4.4. Method 4. Index Theory Approach. This method is based on the Poincare-Hopf index theorem found in differential topology (Guillemin and Pollak [42]). Similar to the univalence mapping approach, it requires a certain sign from the Hessian, but this requirement need hold only at the equilibrium point.

Theorem 7. *Suppose the strategy space of the game is convex and all payoff functions are quasiconcave. Then, if* $(-1)^n|H|$ *is positive whenever* $\partial\pi_i/\partial x_i=0$*, all* i*, there is a unique NE.*

Observe that the condition $(-1)^n|H|$ is trivially satisfied if $|H|$ is negative definite, which is implied by the condition (2) of contraction mapping, i.e., this method is also somewhat weaker than the contraction mapping argument. Moreover, the index theory condition need only hold at the equilibrium. This makes it the most general, but also the hardest to apply. To gain some intuition about why the index theory method works, consider the two-player game. The condition of Theorem 7 simplifies to

$$\begin{vmatrix}\dfrac{\partial^2\pi_1}{\partial x_1^2} & \dfrac{\partial^2\pi_1}{\partial x_1\partial x_2}\\ \dfrac{\partial^2\pi_2}{\partial x_2\partial x_1} & \dfrac{\partial^2\pi_2}{\partial x_2^2}\end{vmatrix}>0\quad \forall x_1,x_2:\frac{\partial\pi_1}{\partial x_1}=0,\frac{\partial\pi_2}{\partial x_2}=0,$$

which can be interpreted as meaning the multiplication of the slopes of best response functions should not exceed one at the equilibrium:

$$\frac{\partial f_1}{\partial x_2}\frac{\partial f_2}{\partial x_1}<1\quad \text{at } x_1^*,x_2^*. \tag{5}$$

As with the contraction mapping approach, with two players, the Theorem becomes easy to visualize. Suppose we have found best response functions $x_1^*=f_1(x_2)$ and $x_2^*=f_2(x_1)$ as in Figure 2. Find an inverse function $x_2=f_1^{-1}(x_1)$ and construct an auxiliary function $g(x_1)=f_1^{-1}(x_1)-f_2(x_1)$ that measures the distance between two best responses. It remains to show that $g(x_1)$ crosses zero only once because this would directly imply a single crossing point of $f_1(x_1)$ and $f_2(x_2)$. Suppose we could show that every time $g(x_1)$ crosses zero, it does so *from below*. If that is the case, we are assured there is only a single crossing: It is impossible for a continuous function to cross zero more than once from below because it would also have to cross zero from above somewhere. It can be shown that the function $g(x_1)$ crosses zero only from below if the slope of $g(x_1)$ *at the crossing point* is positive as follows

$$\frac{\partial g(x_1)}{\partial x_1}=\frac{\partial f_1^{-1}(x_1)}{\partial x_1}-\frac{\partial f_2(x_1)}{\partial x_1}=\frac{1}{\partial f_2(x_2)/\partial x_2}-\frac{\partial f_2(x_1)}{\partial x_1}>0,$$

which holds if (5) holds. Hence, in a two-player game condition, (5) is sufficient for the uniqueness of the NE. Note that condition (5) trivially holds in the newsvendor game because each slope is less than one, and, hence, the multiplication of slopes is less than one as well *everywhere*. Index theory has been used by Netessine and Rudi [71] to show uniqueness of the NE in a retailer-wholesaler game when both parties stock inventory and sell directly to consumers and by Cachon and Kok [21] and Cachon and Zipkin [24].

2.5. Multiple Equilibria

Many games are just not blessed with a unique equilibrium. The next best situation is to have a few equilibria. The worst situation is either to have an infinite number of equilibria or no equilibrium at all. The obvious problem with multiple equilibria is that the players may not know which equilibrium will prevail. Hence, it is entirely possible that a nonequilibrium outcome results because one player plays one equilibrium strategy while a second player chooses a strategy associated with another equilibrium. However, if a game is repeated, then it is possible that the players eventually find themselves in one particular equilibrium. Furthermore, that equilibrium may not be the most desirable one.

If one does not want to acknowledge the possibility of multiple outcomes due to multiple equilibria, one could argue that one equilibrium is more reasonable than the others. For example, there may exist only one symmetric equilibrium, and one may be willing to argue that a symmetric equilibrium is more focal than an asymmetric equilibrium. (See Mahajan and van Ryzin [58] for an example). In addition, it is generally not too difficult to demonstrate the uniqueness of a symmetric equilibrium. If the players have unidimensional strategies, then the system of n first-order conditions reduces to a single equation, and one need only show that there is a unique solution to that equation to prove the symmetric equilibrium is unique. If the players have m-dimensional strategies, $m > 1$, then finding a symmetric equilibrium reduces to determining whether a system of m equations has a unique solution (easier than the original system, but still challenging).

An alternative method to rule out some equilibria is to focus only on the Pareto optimal equilibrium, of which there may be only one. For example, in supermodular games, the equilibria are Pareto rankable under an additional condition that each player's objective function is increasing in other players' strategies, i.e., there is a most preferred equilibrium by every player and a least preferred equilibrium by every player. (See Wang and Gerchak [104] for an example.) However, experimental evidence exists that suggests players do not necessarily gravitate to the Pareto optimal equilibrium as is demonstrated by Cachon and Camerer [19]. Hence, caution is warranted with this argument.

2.6. Comparative Statics in Games

In GT models, just as in the noncompetitive SCM models, many of the managerial insights and results are obtained through comparative statics, such as monotonicity of the optimal decisions w.r.t. some parameter of the game.

2.6.1. The Implicit Functions Theorem Approach. This approach works for both GT and single decision-maker applications, as will become evident from the statement of the next theorem.

Theorem 8. *Consider the system of equations*

$$\frac{\partial \pi_i(x_1, \ldots, x_n, a)}{\partial x_i} = 0, \quad i = 1, \ldots, n,$$

defining $x_1^*, \ldots, x_n^*$ as implicit functions of parameter a. If all derivatives are continuous functions and the Hessian (3) evaluated at $x_1^*, \ldots, x_n^*$ is nonzero, then the function $x^*(a)$: $R^1 \to R^n$ is continuous on a ball around x^* and its derivatives are found as follows:

$$\begin{vmatrix} \frac{\partial x_1^*}{\partial a} \\ \frac{\partial x_2^*}{\partial a} \\ \cdots \\ \frac{\partial x_n^*}{\partial a} \end{vmatrix} = - \begin{vmatrix} \frac{\partial^2 \pi_1}{\partial x_1^2} & \frac{\partial^2 \pi_1}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 \pi_1}{\partial x_1 \partial x_n} \\ \frac{\partial^2 \pi_2}{\partial x_2 \partial x_1} & \frac{\partial^2 \pi_2}{\partial x_2^2} & \cdots & \frac{\partial^2 \pi_1}{\partial x_2 \partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 \pi_n}{\partial x_n \partial x_1} & \frac{\partial^2 \pi_n}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 \pi_n}{\partial x_n^2} \end{vmatrix}^{-1} \begin{vmatrix} \frac{\partial \pi_1}{\partial x_1 \partial a} \\ \frac{\partial \pi_1}{\partial x_2 \partial a} \\ \cdots \\ \frac{\partial \pi_1}{\partial x_n \partial a} \end{vmatrix}. \tag{6}$$

Because the IFT is covered in detail in many nonlinear programming books and its application to the GT problems is essentially the same, we do not delve further into this matter. In many practical problems, if $|H| \neq 0$, then it is instrumental to multiply both sides of the expression (6) by H^{-1}. That is justified because the Hessian is assumed to have a nonzero determinant to avoid the cumbersome task of inverting the matrix. The resulting expression is a system of n linear equations, which have a closed-form solution. See Netessine and Rudi [71] for such an application of the IFT in a two-player game and Bernstein and Federgruen [8] in n-player games.

The solution to (6) in the case of two players is

$$\frac{\partial x_1^*}{\partial a} = - \frac{\frac{\partial^2 \pi_1}{\partial x_1 \partial a} \frac{\partial^2 \pi_2}{\partial x_2^2} - \frac{\partial^2 \pi_1}{\partial x_1 \partial x_2} \frac{\partial^2 \pi_2}{\partial x_2 \partial a}}{|H|}, \tag{7}$$

$$\frac{\partial x_2^*}{\partial a} = - \frac{\frac{\partial^2 \pi_1}{\partial x_1^2} \frac{\partial^2 \pi_2}{\partial x_2 \partial a} - \frac{\partial^2 \pi_1}{\partial x_1 \partial a} \frac{\partial^2 \pi_2}{\partial x_2 \partial x_1}}{|H|}. \tag{8}$$

Using our newsvendor game as an example, suppose we would like to analyze sensitivity of the equilibrium solution to changes in r_1 so let $a = r_1$. Notice that $\partial^2 \pi_2 / \partial Q_2 \partial r_1$ and also that the determinant of the Hessian is positive. Both expressions in the numerator of (7) are positive as well, so that $\partial Q_1^* / \partial r_1 > 0$. Further, the numerator of (8) is negative, so that $\partial Q_2^* / \partial r_1 < 0$. Both results are intuitive.

Solving a system of n equations analytically is generally cumbersome, and one may have to use Kramer's rule or analyze an inverse of H instead, see Bernstein and Federgruen [8] for an example. The only way to avoid this complication is to employ supermodular games as described below. However, the IFT method has an advantage that is not enjoyed by supermodular games: It can handle constraints of any form. That is, any constraint on the players' strategy spaces of the form $g_i(x_i) \leq 0$ or $g_i(x_i) = 0$ can be added to the objective function by forming a Lagrangian:

$$L_i(x_1, \ldots, x_n, \lambda_i) = \pi_i(x_1, \ldots, x_n) - \lambda_i g_i(x_i).$$

All analysis can then be carried through the same way as before with the only addition being that the Lagrange multiplier λ_i becomes a decision variable. For example, let us assume in the newsvendor game that the two competing firms stock inventory at a warehouse. Further, the amount of space available to each company is a function of the total warehouse capacity C, e.g., $g_i(Q_i) \leq C$. We can construct a new game in which each retailer solves the following problem:

$$\max_{Q_i \in \{g_i(Q_i) \leq C\}} E_D[r_i \min(D_i + (D_j - Q_j)^+, Q_i) - c_i Q_i], \quad i = 1, 2.$$

Introduce two Lagrange multipliers, λ_i, $i = 1, 2$ and rewrite the objective functions as

$$\max_{Q_i,\lambda_i} L(Q_i, \lambda_i, Q_j) = E_D[r_i \min(D_i + (D_j - Q_j)^+, Q_i) - c_i Q_i - \lambda_i(g_i(Q_i) - C)].$$

The resulting four optimality conditions can be analyzed using the IFT the same way as has been demonstrated previously.

2.6.2. Supermodular Games Approach. In some situations, supermodular games provide a more convenient tool for comparative statics.

Theorem 9. *Consider a collection of supermodular games on R^n parameterized by a parameter a. Further, suppose $\partial^2\pi_i/\partial x_i \partial a \geq 0$ for all i. Then, the largest and the smallest equilibria are increasing in a.*

Roughly speaking, a sufficient condition for monotone comparative statics is supermodularity of players' payoffs in strategies *and* a parameter. Note that, if there are multiple equilibria, we cannot claim that every equilibrium is monotone in a; rather, a set of all equilibria is monotone in the sense of Theorem 9. A convenient way to think about the last Theorem is through the augmented Hessian:

$$\begin{vmatrix} \frac{\partial^2\pi_1}{\partial x_1^2} & \frac{\partial^2\pi_1}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2\pi_1}{\partial x_1 \partial x_n} & \frac{\partial^2\pi_1}{\partial x_1 \partial a} \\ \frac{\partial^2\pi_2}{\partial x_2 \partial x_1} & \frac{\partial^2\pi_2}{\partial x_2^2} & \cdots & \frac{\partial^2\pi_1}{\partial x_2 \partial x_n} & \frac{\partial^2\pi_1}{\partial x_2 \partial a} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2\pi_n}{\partial x_n \partial x_1} & \frac{\partial^2\pi_n}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2\pi_n}{\partial x_n^2} & \frac{\partial^2\pi_n}{\partial x_n \partial a} \\ \frac{\partial^2\pi_1}{\partial x_1 \partial a} & \frac{\partial^2\pi_1}{\partial x_2 \partial a} & \cdots & \frac{\partial^2\pi_n}{\partial x_n \partial a} & \frac{\partial^2\pi_n}{\partial a^2} \end{vmatrix}.$$

Roughly, if all *off-diagonal* elements of this matrix are positive, then the monotonicity result holds (signs of diagonal elements do not matter and, hence, concavity is not required). To apply this result to competing newsvendors, we will analyze sensitivity of equilibrium inventories (Q_i^*, Q_j^*) to r_i. First, transform the game to strategies (Q_i, y) so that the game is supermodular and find cross-partial derivatives

$$\frac{\partial^2\pi_i}{\partial Q_i \partial r_i} = \Pr(D_i + (D_j - Q_j)^+ > Q_i) \geq 0,$$

$$\frac{\partial\pi_j}{\partial y \partial r_i} = 0 \geq 0,$$

so that (Q_i^*, y^*) are both increasing in r_i, or Q_i^* is increasing and Q_j^* is decreasing in r_i just as we have already established using the IFT.

The simplicity of the argument (once supermodular games are defined) as compared to the machinery required to derive the same result using the IFT is striking. Such simplicity has attracted much attention in SCM and has resulted in extensive applications of supermodular games. Examples include Cachon [16], Corbett and DeCroix [27], and Netessine and Rudi [71] to name just a few. There is, however, an important limitation to the use of Theorem 9: It cannot handle many constraints as IFT can. Namely, the decision space must be a lattice to apply supermodularity, i.e., it must include its coordinatewise maximum and minimum. Hence, a constraint of the form $x_i \leq b$ can be handled, but a constraint $x_i + x_j \leq b$ cannot because points $(x_i, x_j) = (b, 0)$ and $(x_i, x_j) = (0, b)$ are within the constraint but the coordinatewise maximum of these two points (b, b) is not. Notice that to avoid dealing with this issue in detail, we stated in the theorems that the strategy space should all be R^n. Because many SCM applications have constraints on the players' strategies, supermodularity must be applied with care.

3. Dynamic Games

While many SCM models are static—including all newsvendor-based models—a significant portion of the SCM literature is devoted to dynamic models in which decisions are made over time. In most cases, the solution concept for these games is similar to the backward induction used when solving dynamic programming problems. There are, however, important differences, as will be clear from the discussion of repeated games. As with dynamic programming problems, we continue to focus on the games of complete information, i.e., at each move in the game all players know the full history of play.

3.1. Sequential Moves: Stackelberg Equilibrium Concept

The simplest possible dynamic game was introduced by von Stackelberg [103]. In a Stackelberg duopoly model, player 1—the Stackelberg leader—chooses a strategy first, and then player 2—the Stackelberg follower—observes this decision and makes his own strategy choice. Because in many SCM models the upstream firm—e.g., the wholesaler—possesses certain power over the typically smaller downstream firm—e.g., the retailer—the Stackelberg equilibrium concept has found many applications in SCM literature. We do not address the issues of who should be the leader and who should be the follower; see Chapter 11 in Simchi-Levi et al. [88].

To find an equilibrium of a Stackelberg game, which often is called the Stackelberg equilibrium, we need to solve a dynamic multiperiod problem via backward induction. We will focus on a two-period problem for analytical convenience. First, find the solution $x_2^*(x_1)$ for the second player as a response to any decision made by the first player:

$$x_2^*(x_1): \quad \frac{\partial \pi_2(x_2, x_1)}{\partial x_2} = 0.$$

Next, find the solution for the first player anticipating the response by the second player:

$$\frac{d\pi_1(x_1, x_2^*(x_1))}{dx_1} = \frac{\partial \pi_1(x_1, x_2^*)}{\partial x_1} + \frac{\partial \pi_1(x_1, x_2)}{\partial x_2}\frac{\partial x_2^*}{\partial x_1} = 0.$$

Intuitively, the first player chooses the best possible point on the second player's best response function. Clearly, the first player can choose an NE, so the leader is always at least as well off as he would be in NE. Hence, if a player were allowed to choose between making moves simultaneously or being a leader in a game with complete information, he would always prefer to be the leader. However, if new information is revealed after the leader makes a play, then it is not always advantageous to be the leader.

Whether the follower is better off in the Stackelberg or simultaneous move game depends on the specific problem setting. See Netessine and Rudi [70] for examples of both situations and comparative analysis of Stackelberg versus NE; see also Wang and Gerchak [104] for a comparison between the leader versus follower roles in a decentralized assembly model. For example, consider the newsvendor game with sequential moves. The best response function for the second player remains the same as in the simultaneous move game:

$$Q_2^*(Q_1) = F^{-1}_{D_2+(D_1-Q_1)^+}\left(\frac{r_2 - c_2}{r_2}\right).$$

For the leader, the optimality condition is

$$\begin{aligned}\frac{d\pi_1(Q_1, Q_2^*(Q_1))}{dQ_1} &= r_1 \Pr(D_1 + (D_2 - Q_2)^+ > Q_1) - c_1 \\ &\quad - r_1 \Pr(D_1 + (D_2 - Q_2)^+ < Q_1, D_2 > Q_2)\frac{\partial Q_2^*}{\partial Q_1} \\ &= 0,\end{aligned}$$

where $\partial Q_2^*/\partial Q_1$ is the slope of the best response function found in (1). Existence of a Stackelberg equilibrium is easy to demonstrate given the continuous payoff functions. However, uniqueness may be considerably harder to demonstrate. A sufficient condition is quasiconcavity of the leader's profit function, $\pi_1(x_1, x_2^*(x_1))$. In the newsvendor game example, this implies the necessity of finding derivatives of the density function of the demand distribution, as is typical for many problems involving uncertainty. In stochastic models, this is feasible with certain restrictions on the demand distribution. See Lariviere and Porteus [53] for an example with a supplier that establishes the wholesale price and a newsvendor that then chooses an order quantity and Cachon [18] for the reverse scenario in which a retailer sets the wholesale price and buys from a newsvendor supplier. See Netessine and Rudi [70] for a Stackelberg game with a wholesaler choosing a stocking quantity and the retailer deciding on promotional effort. One can further extend the Stackelberg equilibrium concept into multiple periods; see Erhun et al. [34] and Anand et al. [1] for examples.

3.2. Simultaneous Moves: Repeated and Stochastic Games

A different type of dynamic game arises when both players take actions in multiple periods. Because inventory models used in SCM literature often involve inventory replenishment decisions that are made over and over again, multiperiod games should be a logical extension of these inventory models. Two major types of multiple-period games exist: without and with time dependence.

In the multiperiod game without time dependence, the exact same game is played over and over again, hence, the term *repeated* games. The strategy for each player is now a sequence of actions taken in all periods. Consider one repeated game version of the newsvendor game in which the newsvendor chooses a stocking quantity at the start of each period, demand is realized, and then leftover inventory is salvaged. In this case, there are no links between successive periods other than the players' memories about actions taken in all the previous periods. Although repeated games have been extensively analyzed in economics literature, it is awkward in an SCM setting to assume that nothing links successive games; typically, in SCM, there is some transfer of inventory and/or backorders between periods. As a result, repeated games thus far have not found many applications in the SCM literature. Exceptions are Debo [28], Ren et al. [79], and Taylor and Plambeck [94] in which reputational effects are explored as means of supply chain coordination in place of the formal contracts.

A fascinating feature of repeated games is that the set of equilibria is much larger than the set of equilibria in a static game, and may include equilibria that are not possible in the static game. At first, one may assume that the equilibrium of the repeated game would be to play the same static NE strategy in each period. This is, indeed, an equilibrium, but only one of many. Because in repeated games the players are able to condition their behavior on the observed actions in the previous periods, they may employ so-called *trigger strategies*: The player will choose one strategy until the opponent changes his play, at which point the first player will change the strategy. This *threat* of reverting to a different strategy may even induce players to achieve the best possible outcome, i.e., the centralized solution, which is called an *implicit collusion*. Many such threats are, however, noncredible in the sense that once a part of the game has been played, such a strategy is not an equilibrium anymore for the remainder of the game, as is the case in our example in Figure 1. To separate out credible threats from noncredible, Selten [82] introduced the notion of a *subgame-perfect equilibrium*. See Hall and Porteus [43] and Van Mieghem and Dada [98] for solutions involving subgame-perfect equilibria in dynamic games.

Subgame-perfect equilibria reduce the equilibrium set somewhat. However, infinitely repeated games are still particularly troublesome in terms of multiplicity of equilibria. The

famous Folk theorem[5] proves that any convex combination of the feasible payoffs is attainable in the infinitely repeated game as an equilibrium, implying that "virtually anything" is an equilibrium outcome.[6] See Debo [28] for the analysis of a repeated game between the wholesaler setting the wholesale price and the newsvendor setting the stocking quantity.

In time-dependent multiperiod games, players' payoffs in each period depend on the actions in the previous as well as current periods. Typically, the payoff structure does not change from period to period (so called stationary payoffs). Clearly, such setup closely resembles multiperiod inventory models in which time periods are connected through the transfer of inventories and backlogs. Due to this similarity, time-dependent games have found applications in SCM literature. We will only discuss one type of time-dependent multiperiod games, *stochastic games* or *Markov games*, due to their wide applicability in SCM. See also Majumder and Groenevelt [61] for the analysis of deterministic time-dependent multiperiod games in reverse logistics supply chains. Stochastic games were developed by Shapley [84] and later by Heyman and Sobel [45], Kirman and Sobel [48], and Sobel [89]. The theory of stochastic games is also extensively covered in Filar and Vrieze [36].

The setup of the stochastic game is essentially a combination of a static game and a Markov decisions process: In addition to the set of players with strategies—which is now a vector of strategies, one for each period, and payoffs—we have a set of states and a transition mechanism $p(s'|s,x)$, probability that we transition from state s to state s' given action x. Transition probabilities are typically defined through random demand occurring in each period. The difficulties inherent in considering nonstationary inventory models are passed over to the game-theoretic extensions of these models, therefore, a standard simplifying assumption is that demands are independent and identical across periods. When only a single decision maker is involved, such an assumption leads to a unique stationary solution (e.g., stationary inventory policy of some form: order-up-to, S-s, etc.). In a GT setting, however, things get more complicated; just as in the repeated games described above, nonstationary equilibria, e.g., trigger strategies, are possible. A standard approach is to consider just one class of equilibria—e.g., stationary—because nonstationary policies are hard to implement in practice and they are not always intuitively appealing. Hence, with the assumption that the policy is stationary, the stochastic game reduces to an equivalent static game, and equilibrium is found as a sequence of NE in an appropriately modified single-period game. Another approach is to focus on "Markov" or "state-space" strategies in which the past influences the future through the state variables but not through the history of the play. A related equilibrium concept is that of *Markov perfect equilibrium* (MPE), which is simply a profile of Markov strategies that yields a Nash equilibrium in every subgame. The concept of MPE is discussed in Fudenberg and Tirole [38], Chapter 13. See also Tayur and Yang [95] for the application of this concept.

To illustrate, consider an infinite-horizon variant of the newsvendor game with lost sales in each period and inventory carry-over to the subsequent period; see Netessine et al. [74] for complete analysis. The solution to this problem in a noncompetitive setting is an order-up-to policy. In addition to unit-revenue r and unit-cost c, we introduce inventory holding cost h incurred by a unit carried over to the next period and a discount factor β. Also, denote by x_i^t the inventory position at the beginning of the period and by y_i^t the order-up-to quantity. Then, the infinite-horizon profit of each player is

$$\pi_i(x^1) = E\sum_{t=1}^{\infty} \beta_i^{t-1}[r_i \min(y_i^t, D_i^t + (D_j^t - y_j^t)^+) - h_i(y_i^t - D_i^t - (D_j^t - y_j^t)^+)^+ - c_i Q_i^t],$$

[5]The name is due to the fact that its source is unknown and dates back to 1960; Friedman [37] was one of the first to treat Folk theorem in detail.

[6]A condition needed to insure attainability of an equilibrium solution is that the discount factor is large enough. The discount factor also affects effectiveness of trigger and many other strategies.

with the inventory transition equation

$$x_i^{t+1} = (y_i^t - D_i^t - (D_j^t - y_j^t)^+)^+.$$

Using the standard manipulations from Heyman and Sobel [45], this objective function can be converted to

$$\pi_i(x^1) = c_i x_i^1 + \sum_{t=1}^{\infty} \beta_i^{t-1} G_i^t(y_i^t), \quad i = 1, 2,$$

where $G_i^t(y_i^t)$ is a single-period objective function

$$G_i^t(y_i^t) = E[(r_i - c_i)(D_i^t + (D_j^t - y_j^t)^+) - (r_i - c_i)(D_i^t + (D_j^t - y_j^t)^+ - y_i^t)^+ - (h_i + c_i(1 - \beta_i))(y_i^t - D_i^t - (D_j^t - y_j^t)^+)^+], \quad i = 1, 2,\ t = 1, 2, \ldots .$$

Assuming demand is stationary and independently distributed across periods $D_i = D_i^t$, we further obtain that $G_i^t(y_i^t) = G_i(y_i^t)$ because the single-period game is the same in each period. By restricting consideration to the stationary inventory policy $y_i = y_i^t$, $t = 1, 2, \ldots$, we can find the solution to the multiperiod game as a sequence of the solutions to a single-period game $G_i(y_i)$, which is

$$y_i^* = F^{-1}_{D_i + (D_j - y_j^*)^+}\left(\frac{r_i - c_i}{r_i + h_i - c_i\beta_i}\right), \quad i = 1, 2.$$

With the assumption that the equilibrium is stationary, one could argue that stochastic games are no different from static games; except for a small change in the right-hand side reflecting inventory carry-over and holding costs, the solution is essentially the same. However, more elaborate models capture some effects that are not present in static games but can be envisioned in stochastic games. For example, if we were to introduce backlogging in the above model, a couple of interesting situations would arise: A customer may backlog the product with either the first or with the second competitor he visits if both are out of stock. These options introduce the behavior that is observed in practice but cannot be modeled within the static game (see Netessine et al. [74] for detailed analysis) because firms' inventory decisions affect their demand in the future. Among other applications of stochastic games are papers by Cachon and Zipkin [24] analyzing a two-echelon game with the wholesaler and the retailer making stocking decisions, Bernstein and Federgruen [10] analyzing price and service competition, Netessine and Rudi [70] analyzing the game with the retailer exerting sales effort and the wholesaler stocking the inventory, and Van Mieghem and Dada [98] studying a two-period game with capacity choice in the first period and production decision under the capacity constraint in the second period.

3.3. Differential Games

So far, we have described dynamic games in discrete time, i.e., games involving a sequence of decisions separated in time. *Differential games* provide a natural extension for decisions that have to be made continuously. Because many SC models rely on continuous-time processes, it is natural to assume that differential games should find a variety of applications in SCM literature. However, most SCM models include stochasticity in one form or another. At the same time, due to the mathematical difficulties inherent in differential games, we are only aware of deterministic differential GT models in SCM. Although theory for stochastic differential games does exist, applications are quite limited (Basar and Olsder [6]). Marketing and economics have been far more successful in applying differential games because deterministic models are standard in these areas. Hence, we will only briefly outline some new concepts necessary to understand the theory of differential games.

The following is a simple example of a differential game taken from Kamien and Schwartz [47]. Suppose two players indexed by $i = 1, 2$ are engaged in production and sales of the same product. Firms choose production levels $u_i(t)$ at any moment of time and incur total cost $C_i(u_i) = cu_i + u_i^2/2$. The price in the market is determined as per Cournot competition. Typically, this would mean that $p(t) = a - u_1(t) - u_2(t)$. However, the twist in this problem is that if the production level is changed, price adjustments are not instantaneous. Namely, there is a parameter s, referred to as the speed of price adjustment, so that the price is adjusted according to the following differential equation:

$$p'(t) = s[a - u_1(t) - u_2(t) - p(t)], \quad p(0) = p_0.$$

Finally, each firm maximizes discounted total profit

$$\pi_i = \int_0^\infty e^{-rt}(p(t)u_i(t) - C_i(u_i(t)))\,dt, \quad i = 1, 2.$$

The standard tools needed to analyze differential games are the calculus of variations or optimal control theory (Kamien and Schwartz [47]). In a standard optimal control problem, a single decision maker sets the control variable that affects the state of the system. In contrast, in differential games, several players select control variables that may affect a common state variable and/or payoffs of all players. Hence, differential games can be looked at as a natural extension of the optimal control theory. In this section, we will consider two distinct types of player strategies: *open loop* and *closed loop*, which is also sometimes called *feedback*. In the open-loop strategy, the players select their decisions or control variables once at the beginning of the game and do not change them, so that the control variables are only functions of time and do not depend on the other players' strategies. Open-loop strategies are simpler in that they can be found through the straightforward application of optimal control that makes them quite popular. Unfortunately, an open-loop strategy may not be subgame perfect. On the contrary, in a closed-loop strategy, the player bases his strategy on current time and the states of both players' systems. Hence, feedback strategies are subgame perfect: If the game is stopped at any time, for the remainder of the game, the same feedback strategy will be optimal, which is consistent with the solution to the dynamic programming problems that we employed in the stochastic games section. The concept of a feedback strategy is more satisfying, but is also more difficult to analyze. In general, optimal open-loop and feedback strategies differ, but they may coincide in some games.

Because it is hard to apply differential game theory in stochastic problems, we cannot utilize the competitive newsvendor problem to illustrate the analysis. Moreover, the analysis of even the most trivial differential game is somewhat involved mathematically, so we will limit our survey to stating and contrasting optimality conditions in the cases of open-loop and closed-loop NE. Stackelberg equilibrium models do exist in differential games as well but are rarer (Basar and Olsder [6]). Due to mathematical complexity, games with more than two players are rarely analyzed. In a differential game with two players, each player is endowed with a control $u_i(t)$ that the player uses to maximize the objective function π_i

$$\max_{u_i(t)} \pi_i(u_i, u_j) = \max_{u_i(t)} \int_0^T f_i(t, x_i(t), x_j(t), u_i(t), u_j(t))\,dt,$$

where $x_i(t)$ is a state variable describing the state of the system. The state of the system evolves according to the differential equation

$$x_i'(t) = g_i(t, x_i(t), x_j(t), u_i(t), u_j(t)),$$

which is the analog of the inventory transition equation in the multiperiod newsvendor problem. Finally, there are initial conditions $x_i(0) = x_{i0}$.

The open-loop strategy implies that each player's control is only a function of time, $u_i = u_i(t)$. A feedback strategy implies that each players' control is also a function of state variables, $u_i = u_i(t, x_i(t), x_j(t))$. As in the static games, NE is obtained as a fixed point of the best response mapping by simultaneously solving a system of first-order optimality conditions for the players. Recall that to find the optimal control, we first need to form a Hamiltonian. If we were to solve two individual noncompetitive optimization problems, the Hamiltonians would be $H_i = f_i + \lambda_i g_i$, $i = 1, 2$, where $\lambda_i(t)$ is an adjoint multiplier. However, with two players, we also have to account for the state variable of the opponent so that the Hamiltonian becomes

$$H_i = f_i + \lambda_i^1 g_i + \lambda_i^2 g_j, \quad i, j = 1, 2.$$

To obtain the necessary conditions for the open-loop NE, we simply use the standard necessary conditions for any optimal control problem:

$$\frac{\partial H_1}{\partial u_1} = 0, \quad \frac{\partial H_2}{\partial u_2} = 0, \tag{9}$$

$$\frac{\partial \lambda_1^1}{\partial t} = -\frac{\partial H_1}{\partial x_1}, \quad \frac{\partial \lambda_1^2}{\partial t} = -\frac{\partial H_1}{\partial x_2}, \tag{10}$$

$$\frac{\partial \lambda_2^1}{\partial t} = -\frac{\partial H_2}{\partial x_2}, \quad \frac{\partial \lambda_2^2}{\partial t} = -\frac{\partial H_2}{\partial x_1}. \tag{11}$$

For the feedback equilibrium, the Hamiltonian is the same as for the open-loop strategy. However, the necessary conditions are somewhat different:

$$\frac{\partial H_1}{\partial u_1} = 0, \quad \frac{\partial H_2}{\partial u_2} = 0, \tag{12}$$

$$\frac{\partial \lambda_1^1}{\partial t} = -\frac{\partial H_1}{\partial x_1} - \frac{\partial H_1}{\partial u_2}\frac{\partial u_2^*}{\partial x_1}, \quad \frac{\partial \lambda_1^2}{\partial t} = -\frac{\partial H_1}{\partial x_2} - \frac{\partial H_1}{\partial u_2}\frac{\partial u_2^*}{\partial x_2}, \tag{13}$$

$$\frac{\partial \lambda_2^1}{\partial t} = -\frac{\partial H_2}{\partial x_2} - \frac{\partial H_2}{\partial u_1}\frac{\partial u_1^*}{\partial x_2}, \quad \frac{\partial \lambda_2^2}{\partial t} = -\frac{\partial H_2}{\partial x_1} - \frac{\partial H_2}{\partial u_1}\frac{\partial u_1^*}{\partial x_1}. \tag{14}$$

Notice that the difference is captured by an extra term on the right when we compare (10) and (13) or (11) and (14). The difference is because the optimal control of each player under the feedback strategy depends on $x_i(t)$, $i = 1, 2$. Hence, when differentiating the Hamiltonian to obtain Equations (13) and (14), we have to account for such dependence (note also that two terms disappear when we use (12) to simplify).

As we mentioned earlier, there are numerous applications of differential games in economics and marketing, especially in the area of dynamic pricing, see Eliashberg and Jeuland [32]. Desai [30, 31] and Eliashberg and Steinberg [33] use the open-loop Stackelberg equilibrium concept in a marketing-production game with the manufacturer and the distributor. Gaimon [39] uses both open and closed-loop NE concepts in a game with two competing firms choosing prices and production capacity when the new technology reduces firms' costs. Mukhopadhyay and Kouvelis [64] consider a duopoly with firms competing on prices and quality of design and derive open- and closed-loop NE.

4. Cooperative Games

The subject of cooperative games first appeared in the seminal work of von Neumann and Morgenstern [102]. However, for a long time, cooperative game theory did not enjoy as much attention in the economics literature as noncooperative GT. Papers employing cooperative GT to study SCM had been scarce, but are becoming more popular. This trend is probably due to the prevalence of bargaining and negotiations in SC relationships.

Cooperative GT involves a major shift in paradigms as compared to noncooperative GT: The former focuses on the outcome of the game in terms of the value created through cooperation of a subset of players but does not specify the actions that each player will take, while the latter is more concerned with the specific actions of the players. Hence, cooperative GT allows us to model outcomes of complex business processes that otherwise might be too difficult to describe, e.g., negotiations, and answers more general questions, e.g., how well is the firm positioned against competition (Brandenburger and Stuart [14]). However, there are also limitations to cooperative GT, as we will later discuss.

In what follows, we will cover transferable utility cooperative games (players can share utility via side payments) and two solution concepts: The core of the game and the Shapley value, and also biform games that have found several applications in SCM. Not covered are alternative concepts of value, e.g., nucleous and the σ-value, and games with nontransferable utility that have not yet found application in SCM. Material in this section is based mainly on Moulin [63] and Stuart [91]. Perhaps the first paper employing cooperative games in SCM is Wang and Parlar [106] who analyze the newsvendor game with three players, first in a noncooperative setting and then under cooperation with and without transferable utility. See Nagarajan and Sosic [67] for a more detailed review of cooperative games including analysis of the concepts of dynamic coalition formation and farsighted stability—issues that we do not address here.

4.1. Games in Characteristic Form and the Core of the Game

Recall that the noncooperative game consists of a set of players with their strategies and payoff functions. In contrast, the cooperative game (which is also called the game in characteristic form) consists of the set of players N with subsets or *coalitions* $S \subseteq N$ and a *characteristic function* $v(S)$ that specifies a (maximum) value (which we assume is a real number) created by any subset of players in N, i.e., the total pie that members of a coalition can create and divide. The specific actions that players have to take to create this value are not specified: The characteristic function only defines the total value that can be created by utilizing all players' resources. Hence, players are free to form any coalitions beneficial to them, and no player is endowed with power of any sort. Furthermore, the value a coalition creates is independent of the coalitions and actions taken by the noncoalition members. This decoupling of payoffs is natural in political settings (e.g., the majority gets to choose the legislation), but it is far more problematic in competitive markets. For example, in the context of cooperative game theory, the value HP and Compaq can generate by merging is independent of the actions taken by Dell, Gateway, IBM, Ingram Micro, etc.[7]

A frequently used solution concept in cooperative GT is the *core of the game*:

Definition 5. The utility vector $\pi_1, \ldots, \pi_N$ is in the core of the cooperative game if $\forall S \subset N, \sum_{i \in S} \pi_i \geq v(S)$ and $\sum_{i \in N} \pi_i \geq v(N)$.

A utility vector is in the core if the total utility of every possible coalition is at least as large as the coalition's value, i.e., there does not exist a coalition of players that could make all of its members at least as well off and one member strictly better off.

As is true for NE, the core of the game may not exist, i.e., it may be empty, and the core is often not unique. Existence of the core is an important issue because with an empty core, it is difficult to predict what coalitions would form and what value each player would receive. If the core exists, then the core typically specifies a range of utilities that a player can appropriate, i.e., competition alone does not fully determine the players' payoffs. What utility each player will actually receive is undetermined: It may depend on details of the residual bargaining process, a source of criticism of the core concept. (Biform games, described below, provide one possible resolution of this indeterminacy.)

[7] One interpretation of the value function is that it is the minimum value a coalition can guarantee for itself assuming the other players take actions that are most damaging to the coalition. However, that can be criticized as overly conservative.

In terms of specific applications to the SCM, Hartman et al. [44] considered the newsvendor centralization game, i.e., a game in which multiple retailers decide to centralize their inventory and split profits resulting from the benefits of risk pooling. Hartman et al. [44] further show that this game has a nonempty core under certain restrictions on the demand distribution. Muller et al. [65] relax these restrictions and show that the core is always nonempty. Further, Muller et al. [65] give a condition under which the core is a singleton.

4.2. Shapley Value

The concept of the core, though intuitively appealing, also possesses some unsatisfying properties. As we mentioned, the core might be empty or indeterministic.[8] For the same reason it is desirable to have a unique NE in noncooperative games, it is desirable to have a solution concept for cooperative games that results in a unique outcome. Shapley [85] offered an axiomatic approach to a solution concept that is based on three axioms. First, the value of a player should not change due to permutations of players, i.e., only the role of the player matters and not names or indices assigned to players. Second, if a player's added value to the coalition is zero then this player should not get any profit from the coalition, or, in other words, only players generating added value should share the benefits. (A player's added value is the difference between the coalition's value with that player and without that player.) Those axioms are intuitive, but the third is far less so. The third axiom requires additivity of payoffs: If v_1 and v_2 are characteristic functions in any two games, and if q_1 and q_2 are a player's Shapely value in these two games, then the player's Shapely value in the composite game, $v_1 + v_2$, must be $q_1 + q_2$. This is not intuitive because it is not clear what is meant by a composite game. Nevertheless, Shapley [85] demonstrates that there is a unique value for each player, called the Shapley value, that satisfies all three axioms.

Theorem 10. *The Shapley value, π_i, for player i in an N-person noncooperative game with transferable utility is*

$$\pi_i = \sum_{S \subseteq N \setminus i} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!}(v(S \cup \{i\}) - v(S)).$$

The Shapley value assigns to each player his marginal contribution $(v(S \cup \{i\} - v(S))$ when S is a random coalition of agents preceding i and the ordering is drawn at random. To explain further (Myerson [66]), suppose players are picked randomly to enter into a coalition. There are $|N|!$ different orderings for all players, and for any set S that does not contain player i there are $|S|!\,(|N| - |S| - 1)!$ ways to order players so that all players in S are picked ahead of player i. If the orderings are equally likely, there is a probability of $|S|!\,(|N| - |S| - 1)!/|N|!$ that when player i is picked, he will find S players in the coalition already. The marginal contribution of adding player i to coalition S is $(v(S \cup \{i\}) - v(S))$. Hence, the Shapley value is nothing more than a marginal expected contribution of adding player i to the coalition.

Because the Shapley value is unique, it has found numerous applications in economics and political sciences. So far, however, SCM applications are scarce: Except for discussion in Granot and Sosic [41] and analysis in Bartholdi and Kemahlioglu-Ziya [5], we are not aware of any other papers employing the concept of the Shapley value. Although uniqueness of the Shapely value is a convenient feature, caution should surely be taken with Shapley value: The Shapley value need not be in the core; hence, although the Shapely is appealing from the perspective of fairness, it may not be a reasonable prediction of the outcome of a game (i.e., because it is not in the core, there exists some subset of players that can deviate and improve their lots).

[8] Another potential problem is that the core might be very large. However, as Brandenburger and Stuart [15] point out, this may happen for a good reason: To interprete such situations, one can think of competition as not having much force in the game, hence the division of value will largely depend on the intangibles involved.

4.3. Biform Games

From the SCM point of view, cooperative games are somewhat unsatisfactory in that they do not explicitly describe the equilibrium actions taken by the players that is often the key in SC models. *Biform games*, developed by Brandenburger and Stuart [15], compensate to some extent for this shortcoming.

A biform game can be thought of as a noncooperative game with cooperative games as outcomes, and those cooperative games lead to specific payoffs. Similar to the noncooperative game, the biform game has a set of players N, a set of strategies for each player, and also a cost function associated with each strategy (cost function is optional—we include it because most SCM applications of biform games involve cost functions). The game begins by players making choices from among their strategies and incurring costs. After that, a cooperative game occurs in which the characteristic value function depends on the chosen actions. Hopefully, the core of each possible cooperative game is nonempty, but it is also unlikely to be unique. As a result, there is no specific outcome of the cooperative subgame, i.e., it is not immediately clear what value each player can expect. The proposed solution is that each player is assigned a confidence index, $\alpha_i \in [0,1]$, and the α_is are common knowledge. Each player then expects to earn in each possible cooperative game a weighted average of the minimum and maximum values in the core, with α_i being the weight. For example, if $\alpha_i = 0$, then the player earns the minimum value in the core, and if $\alpha_i = 1$, then the player earns the maximum value in the core. Once a specific value is assigned to each player for each cooperative subgame, the first stage noncooperative game can be analyzed just like any other noncooperative game.

Biform games have been successfully adopted in several SCM papers. Anupindi et al. [2] consider a game where multiple retailers stock at their own locations as well as at several centralized warehouses. In the first (noncooperative) stage, retailers make stocking decisions. In the second (cooperative) stage, retailers observe demand and decide how much inventory to transship among locations to better match supply and demand and how to appropriate the resulting additional profits. Anupindi et al. [2] conjecture that a characteristic form of this game has an empty core. However, the biform game has a nonempty core, and they find the allocation of rents based on dual prices that is in the core. Moreover, they find an allocation mechanism in the core that allows them to achieve coordination, i.e., the first-best solution. Granot and Sosic [41] analyze a similar problem but allow retailers to hold back the residual inventory. Their model actually has three stages: Inventory procurement, decision about how much inventory to share with others, and finally the transshipment stage. Plambeck and Taylor [76, 77] analyze two similar games between two firms that have an option of pooling their capacity and investments to maximize the total value. In the first stage, firms choose investment into effort that affects the market size. In the second stage, firms bargain over the division of the market and profits. Stuart [92] analyze biform newsvendor game with endogenous pricing.

5. Signaling, Screening, and Bayesian Games

So far, we have considered only games in which the players are on "equal footing" with respect to information, i.e., each player knows every other player's expected payoff with certainty for any set of chosen actions. However, such ubiquitous knowledge is rarely present in supply chains. One firm may have a better forecast of demand than another firm, or a firm may possess superior information regarding its own costs and operating procedures. Furthermore, a firm may know that another firm may have better information, and, therefore, choose actions that acknowledge this information shortcoming. Fortunately, game theory provides tools to study these rich issues, but, unfortunately, they do add another layer of analytical complexity. This section briefly describes three types of games in which the information structure has a strategic role: Signaling games, screening games, and Bayesian

games. Detailed methods for the analysis of these games are not provided. Instead, a general description is provided along with specific references to supply chain management papers that study these games.

5.1. Signaling Game

In its simplest form, a signaling game has two players, one of which has better information than the other, and it is the player with the better information that makes the first move. For example, Cachon and Lariviere [23] consider a model with one supplier and one manufacturer. The supplier must build capacity for a key component to the manufacturer's product, but the manufacturer has a better demand forecast than the supplier. In an ideal world, the manufacturer would truthfully share her demand forecast with the supplier so that the supplier could build the appropriate amount of capacity. However, the manufacturer always benefits from a larger installed capacity in case demand turns out to be high, but it is the supplier that bears the cost of that capacity. Hence, the manufacturer has an incentive to inflate her forecast to the supplier. The manufacturer's hope is that the supplier actually believes the rosy forecast and builds additional capacity. Unfortunately, the supplier is aware of this incentive to distort the forecast, and, therefore, should view the manufacturer's forecast with skepticism. The key issue is whether there is something the manufacturer should do to make her forecast convincing, i.e., credible.

While the reader should refer to Cachon and Lariviere [23] for the details of the game, some definitions and concepts are needed to continue this discussion. The manufacturer's private information, or type, is her demand forecast. There is a set of possible types that the manufacturer could be, and this set is known to the supplier, i.e., the supplier is aware of the possible forecasts, but is not aware of the manufacturer's actual forecast. Furthermore, at the start of the game, the supplier and the manufacturer know the probability distribution over the set of types. We refer to this probability distribution as the supplier's belief regarding the types. The manufacturer chooses her action first, which, in this case, is a contract offer and a forecast, the supplier updates his belief regarding the manufacturer's type given the observed action, and then the supplier chooses his action, which, in this case, is the amount of capacity to build. If the supplier's belief regarding the manufacturer's type is resolved to a single type after observing the manufacturer's action (i.e., the supplier assigns a 100% probability that the manufacturer is that type and a zero probability that the manufacturer is any other type), then the manufacturer has signaled a type to the supplier. The trick is for the supplier to ensure that the manufacturer has signaled her actual type.

While we are mainly interested in the set of contracts that credibly signal the manufacturer's type, it is worth beginning with the possibility that the manufacturer does not signal her type. In other words, the manufacturer chooses an action such that the action does not provide the supplier with additional information regarding the manufacturer's type. That outcome is called a *pooling equilibrium*, because the different manufacturer types behave in the same way, i.e., the different types are pooled into the same set of actions. As a result, Bayes' rule does not allow the supplier to refine his beliefs regarding the manufacturer's type.

A pooling equilibrium is not desirable from the perspective of supply chain efficiency because the manufacturer's type is not communicated to the supplier. Hence, the supplier does not choose the correct capacity given the manufacturer's actual demand forecast. However, this does not mean that both firms are disappointed with a pooling equilibrium. If the manufacturer's demand forecast is less than average, then that manufacturer is quite happy with the pooling equilibrium because the supplier is likely to build more capacity than he would if he learned the manufacturer's true type. It is the manufacturer with a higher-than-average demand forecast that is disappointed with the pooling equilibrium because then the supplier is likely to underinvest in capacity.

A pooling equilibrium is often supported by the belief that every type will play the pooling equilibrium and any deviation from that play would only be done by a manufacturer with a

low-demand forecast. This belief can prevent the high-demand manufacturer from deviating from the pooling equilibrium: A manufacturer with a high-demand forecast would rather be treated as an average demand manufacturer (the pooling equilibrium) than a low-demand manufacturer (if deviating from the pooling equilibrium). Hence, a pooling equilibrium can indeed be an NE in the sense that no player has a unilateral incentive to deviate given the strategies and beliefs chosen by the other players.

While a pooling equilibrium can meet the criteria of an NE, it nevertheless may not be satisfying. In particular, why should the supplier believe that the manufacturer is a low type if the manufacturer deviates from the pooling equilibrium? Suppose the supplier were to believe a deviating manufacturer has a high-demand forecast. If a high-type manufacturer is better off deviating but a low-type manufacturer is not better off, then only the high-type manufacturer would choose such a deviation. The key part in this condition is that the low type is not better off deviating. In that case, it is not reasonable for the supplier to believe the deviating manufacturer could only be a high type, therefore, the supplier should adjust his belief. Furthermore, the high-demand manufacturer should then deviate from the pooling equilibrium, i.e., this reasoning, which is called the intuitive criterion, breaks the pooling equilibrium; see Kreps [49].

The contrast to a pooling equilibrium is a *separating* equilibrium, also called a *signaling* equilibrium. With a separating equilibrium, the different manufacturer types choose different actions, so the supplier is able to perfectly refine his belief regarding the manufacturer's type given the observed action. The key condition for a separating equilibrium is that only one manufacturer type is willing to choose the action designated for that type. If there is a continuum of manufacturer types, then it is quite challenging to obtain a separating equilibrium: It is difficult to separate two manufacturers that have nearly identical types. However, separating equilibria are more likely to exist if there is a finite number of discrete types.

There are two main issues with respect to separating equilibria: What actions lead to separating equilibrium, and does the manufacturer incur a cost to signal, i.e., is the manufacturer's expected profit in the separating equilibrium lower than what it would be if the manufacturer's type were known to the supplier with certainty? In fact, these two issues are related: An ideal action for a high-demand manufacturer is one that costlessly signals her high-demand forecast. If a costless signal does not exist, then the goal is to seek the lowest-cost signal.

Cachon and Lariviere [23] demonstrate that whether a costless signal exists depends on what commitments the manufacturer can impose on the supplier. For example, suppose the manufacturer dictates to the supplier a particular capacity level in the manufacturer's contract offer. Furthermore, suppose the supplier accepts that contract, and by accepting the contract, the supplier has essentially no choice but to build that level of capacity because the penalty for noncompliance is too severe. They refer to this regime as forced compliance. In that case, there exist many costless signals for the manufacturer. However, if the manufacturer's contract is not iron-clad, so the supplier could potentially deviate—which is referred to as voluntary compliance—then the manufacturer's signaling task becomes more complex.

One solution for a high-demand manufacturer is to give a sufficiently large lump-sum payment to the supplier: The high-demand manufacturer's profit is higher than the low-demand manufacturer's profit, so only a high-demand manufacturer could offer that sum. This has been referred to as signaling by "burning money": Only a firm with a lot of money can afford to burn that much money.

While burning money can work, it is not a smart signal: Burning one unit of income hurts the high-demand manufacturer as much as it hurts the low-demand manufacturer. The signal works only because the high-demand manufacturer has more units to burn. A better signal is a contract offer that is costless to a high-demand manufacturer but expensive to

a low-demand manufacturer. A good example of such a signal is a minimum commitment. A minimum commitment is costly only if realized demand is lower than the commitment, because then the manufacturer is forced to purchase more units than desired. That cost is less likely for a high-demand manufacturer, so, in expectation, a minimum commitment is costlier for a low-demand manufacturer. Interestingly, Cachon and Lariviere [23] show that a manufacturer would never offer a minimum commitment with perfect information, i.e., these contracts may be used in practice solely for the purpose of signaling information.

5.2. Screening

In a screening game, the player that lacks information is the first to move. For example, in the screening game version of the supplier-manufacturer game described by Cachon and Lariviere [23], the supplier makes the contract offer. In fact, the supplier offers a menu of contracts with the intention of getting the manufacturer to reveal her type via the contract selected in the menu. In the economics literature, this is also referred to as *mechanism design*, because the supplier is in charge of designing a mechanism to learn the manufacturer's information. See Porteus and Whang [78] for a screening game that closely resembles this one.

The space of potential contract menus is quite large, so large, that it is not immediately obvious how to begin to find the supplier's optimal menu. For example, how many contracts should be offered, and what form should they take? Furthermore, for any given menu, the supplier needs to infer for each manufacturer type which contract the type will choose. Fortunately, the *revelation principle* (Kreps [49]) provides some guidance.

The revelation principle begins with the presumption that a set of optimal mechanisms exists. Associated with each mechanism is an NE that specifies which contract each manufacturer type chooses and the supplier's action given the chosen contract. With some equilibria, it is possible that some manufacturer type chooses a contract, which is not designated for that type. For example, the supplier intends the low-demand manufacturer to choose one of the menu options, but instead, the high-demand manufacturer chooses that option. Even though this does not seem desirable, it is possible that this mechanism is still optimal in the sense that the supplier can do no better on average. The supplier ultimately cares only about expected profit, not the means by which that profit is achieved. Nevertheless, the revelation principle states that an optimal mechanism that involves deception (the wrong manufacturer chooses a contract) can be replaced by a mechanism that does not involve deception, i.e., there exists an equivalent mechanism that is truth telling. Hence, in the hunt for an optimal mechanism, it is sufficient to consider the set of revealing mechanisms: The menu of contracts is constructed such that each option is designated for a type and that type chooses that option.

Even though an optimal mechanism may exist for the supplier, this does not mean the supplier earns as much profit as he would if he knew the manufacturer's type. The gap between what a manufacturer earns with the menu of contracts and what the same manufacturer would earn if the supplier knew her type is called an information rent. A feature of these mechanisms is that separation of the manufacturer types goes hand in hand with a positive information rent, i.e., a manufacturer's private information allows the manufacturer to keep some rent that the manufacturer would not be able to keep if the supplier knew her type. Hence, even though there may be no cost to information revelation with a signaling game, the same is not true with a screening game.

There have been a number of applications of the revelation principle in the supply chain literature: e.g., Chen [25] studies auction design in the context of supplier procurement contracts; Corbett [26] studies inventory contract design; Baiman et al. [4] study procurement of quality in a supply chain.

5.3. Bayesian Games

With a signaling game or a screening game, actions occur sequentially so information can be revealed through the observation of actions. There also exist games with private information that do not involve signaling or screening. Consider the capacity allocation game studied by Cachon and Lariviere [22]. A single supplier has a finite amount of capacity. There are multiple retailers, and each knows his own demand but not the demand of the other retailers. The supplier announces an allocation rule, the retailers submit their orders, and then the supplier produces and allocates units. If the retailers' total order is less than capacity, then each retailer receives his entire order. If the retailers' total order exceeds capacity, the supplier's allocation rule is implemented to allocate the capacity. The issue is the extent to which the supplier's allocation rule influences the supplier's profit, retailer's profit, and supply chain's profit.

In this setting, the firms with the private information (the retailers) choose their actions simultaneously. Therefore, there is no information exchange among the firms. Even the supplier's capacity is fixed before the game starts, so the supplier is unable to use any information learned from the retailers' orders to choose a capacity. However, it is possible that correlation exists in the retailers' demand information, i.e., if a retailer observes his demand type to be high, then he might assess the other retailers' demand type to be high as well (if there is a positive correlation). Roughly speaking, in a Bayesian game, each player uses Bayes' rule to update his belief regarding the types of the other players. An equilibrium is then a set of strategies for each type that is optimal given the updated beliefs with that type and the actions of all other types. See Fudenberg and Tirole [38] for more information on Bayesian games.

6. Summary and Opportunities

As has been noted in other reviews, operations management has been slow to adopt GT. But because SCM is an ideal candidate for GT applications, we have recently witnessed an explosion of GT papers in SCM. As our survey indicates, most of these papers utilize only a few GT concepts, in particular, the concepts related to noncooperative static games. Some attention has been given to stochastic games, but several other important areas need additional work: Cooperative, repeated, differential, signaling, screening, and Bayesian games.

The relative lack of GT applications in SCM can be partially attributed to the absence of GT courses from the curriculum of most doctoral programs in operations research/management. One of our hopes with this survey is to spur some interest in GT tools by demonstrating that they are intuitive and easy to apply for a person with traditional operations research training.

With the invention of the Internet, certain GT tools have received significant attention: Web auctions gave a boost to auction theory, and numerous websites offer an opportunity to haggle, thus making bargaining theory fashionable. In addition, the advent of relatively cheap information technology has reduced transaction costs and enabled a level of disintermediation that could not be achieved before. Hence, it can only become more important to understand the interactions among independent agents within and across firms. While the application of game theory to supply chain management is still in its infancy, much more progress will soon come.

References

[1] K. Anand, R. Anupindi, and Y. Bassok. Strategic inventories in procurement contracts. Working paper, University of Pennsylvania, 2002.

[2] R. Anupindi, Y. Bassok, and E. Zemel. A general framework for the study of decentralized distribution systems. *Manufacturing and Service Operations Management* 3(4):349–368, 2001.

[3] R. J. Aumann. Acceptable points in general cooperative N-person games. A. W. Tucker and R. D. Luce, eds. *Contributions to the Theory of Games*, Vol. IV. Princeton University Press, Princeton, NJ, 1959.

[4] S. Baiman, S. Netessine, and H. Kunreuther. Procurement in supply chains when the end-product exhibits the weakest link property. Working paper, University of Pennsylvania, 2003.

[5] J. J. Bartholdi, III and E. Kemahlioglu-Ziya. Centralizing inventory in supply chains by using shapley value to allocate the profits. Working paper, University of Pennsylvania, 2005.

[6] T. Basar and G. J. Olsder. *Dynamic Noncooperative Game Theory*. SIAM, Philadelphia, PA, 1995.

[7] F. Bernstein and A. Federgruen. Pricing and replenishment strategies in a distribution system with competing retailers. *Operations Research* 51(3):409–426, 2003.

[8] F. Bernstein and A. Federgruen. Comparative statics, strategic complements and substitute in oligopolies. *Journal of Mathematical Economics* 40(6):713–746, 2004.

[9] F. Bernstein and A. Federgruen. A general equilibrium model for decentralized supply chains with price- and service-competition. *Operations Research* 52(6):868–886, 2004.

[10] F. Bernstein and A. Federgruen. Dynamic inventory and pricing models for competing retailers. *Naval Research Logistics* 51(2):258–274, 2004.

[11] F. Bernstein and A. Federgruen. Decentralized supply chains with competing retailers under Demand Uncertainty. *Management Science* 51(1):18–29, 2005.

[12] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Nashua, NH, 1999.

[13] K. C. Border. *Fixed Point Theorems with Applications to Economics and Game Theory*. Cambridge University Press, Cambridge, MA, 1999.

[14] A. Brandenburger and H. W. Stuart, Jr. Value-based business strategy. *Journal of Economics and Management Strategy* 5(1):5–24, 1996.

[15] A. Brandenburger and H. W. Stuart, Jr. Biform games. *Management Science*. Forthcoming. 2006.

[16] G. P. Cachon. Stock wars: Inventory competition in a two-echelon supply chain. *Operations Research* 49(5):658–674, 2001.

[17] G. P. Cachon. Supply chain coordination with contracts. S. Graves and T. de Kok, eds. *Handbooks in Operations Research and Management Science: Supply Chain Management*. Elsevier, Netherlands, 2002.

[18] G. P. Cachon. The allocation of inventory risk in a supply chain: Push, pull and advanced purchase discount contracts. *Management Science* 50(2):222–238, 2004.

[19] G. P. Cachon and C. Camerer. Loss avoidance and forward induction in coordination games. *Quarterly Journal of Economics* 111(1):165–194, 1996.

[20] G. P. Cachon and P. T. Harker. Competition and outsourcing with scale economies. *Management Science* 48(10):1314–1333, 2002.

[21] G. P. Cachon and G. Kok. How to (and how not to) estimate the salvage value in the newsvendor model. Working paper, University of Pennsylvania, 2002.

[22] G. P. Cachon and M. Lariviere. Capacity choice and allocation: strategic behavior and supply chain performance. *Management Science* 45(8):1091–1108, 1999.

[23] G. P. Cachon and M. Lariviere. Contracting to assure supply: How to share demand forecasts in a supply chain. *Management Science* 47(5):629–646, 2001.

[24] G. P. Cachon and P. H. Zipkin. Competitive and cooperative inventory policies in a two-stage supply chain. *Management Science* 45(7):936–953, 1999.

[25] F. Chen. Auctioning supply contracts. Working paper, Columbia University, New York, 2001.

[26] C. J. Corbett. Stochastic inventory systems in a supply chain with asymmetric information: Cycle stocks, safety stocks, and consignment stock. *Operations Research* 49(4):487–500, 2001.

[27] C. J. Corbett and G. A. DeCroix. Shared-savings contracts for indirect materials in supply chains: Channel profits and environmental impacts. *Management Science* 47(7):881–893, 2001.

[28] L. Debo. Repeatedly selling to an impatient newsvendor when demand fluctuates: A supergame framework for co-operation in a supply chain. Working paper, Carnegie Mellon University, Pittsburgh, PA, 1999.

[29] D. Debreu. A social equilibrium existence theorem. *Proceedings of the National Academy of Sciences of the USA* 38:886–893, 1952.

[30] V. S. Desai. Marketing-production decisions under independent and integrated channel structures. *Annals of Operations Research* 34:275–306, 1992.

[31] V. S. Desai. Interactions between members of a marketing-production channel under seasonal demand. *European Journal of Operational Research* 90(1):115–141, 1996.

[32] J. Eliashberg and A. P. Jeuland. The impact of competitive entry in a developing market upon dynamic pricing strategies. *Marketing Science* 5(1):20–36, 1986.

[33] J. Eliashberg and R. Steinberg. Marketing-production decisions in an industrial channel of distribution. *Management Science* 33(8):981–1000, 1987.

[34] F. Erhun, P. Keskinocak, and S. Tayur. Analysis of capacity reservation and spot purchase under horizontal competition. Working paper, Stanford University, Stanford, CA, 2000.

[35] G. Feichtinger and S. Jorgensen. Differential game models in management science. *European Journal of Operational Research* 14(2):137–155, 1983.

[36] J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer-Verlag, Amsterdam, Netherlands, 1996.

[37] J. W. Friedman. *Game Theory with Applications to Economics*. Oxford University Press, New York, 1986.

[38] D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, Cambridge, MA, 1991.

[39] C. Gaimon. Dynamic game results of the acquisition of new technology. *Operations Research* 37(3):410–425, 1989.

[40] D. Gale and H. Nikaido. The Jacobian matrix and global univalence of mappings. *Mathematische Annalen* 159:81–93, 1965.

[41] D. Granot and G. Sosic. A three-stage model for a decentralized distribution system of retailers. *Operations Research* 51(5):771–784, 2003.

[42] V. Guillemin and A. Pollak. *Differential Topology*. Prentice Hall, Upper Saddle River, NJ, 1974.

[43] J. Hall and E. Porteus. Customer service competition in capacitated systems. *Manufacturing and Service Operations Management* 2(2):144–165, 2000.

[44] B. C. Hartman, M. Dror, and M. Shaked. Cores of inventory centralization games. *Games and Economic Behavior* 31(1):26–49, 2000.

[45] D. P. Heyman and M. J. Sobel. *Stochastic Models in Operations Research*, Vol. II: *Stochastic Optimization*. McGraw-Hill, New York, 1984.

[46] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, MA, 1996.

[47] M. I. Kamien and N. L. Schwartz. *Dynamic Optimization: The Calculus of Variations and Optimal Control in Economics and Management*. North-Holland, Netherlands, 2000.

[48] A. P. Kirman and M. J. Sobel. Dynamic oligopoly with inventories. *Econometrica* 42(2): 279–287, 1974.

[49] D. M. Kreps. *A Course in Microeconomic Theory*. Princeton University Press, Princeton, NJ, 1990.

[50] D. M. Kreps and R. Wilson. Sequential equilibria. *Econometrica* 50(4):863–894, 1982.

[51] H. W. Kuhn. Extensive games and the problem of information. H. W. Kuhn and A. W. Tucker, eds. *Contributions to the Theory of Games*, Vol. II. Princeton University Press, Princeton, NJ, 1953.

[52] R. Lal. Price promotions: Limiting competitive encroachment. *Marketing Science* 9(3): 247–262, 1990.

[53] M. A. Lariviere and E. L. Porteus. Selling to the newsvendor: An analysis of price-only contracts. *Manufacturing and Service Operations Management* 3(4):293–305, 2001.

[54] P. Lederer and L. Li. Pricing, production, scheduling, and delivery-time competition. *Operations Research* 45(3):407–420, 1997.

[55] L. Li and S. Whang. Game theory models in operations management and information systems. K. Chatterjee and W. F. Samuelson, eds. *Game Theory and Business Applications*. Springer, New York, 2001.

[56] S. A. Lippman and K. F. McCardle. The competitive newsboy. *Operations Research* 45(1): 54–65, 1997.

[57] W. F. Lucas. An overview of the mathematical theory of games. *Management Science* 18(5): 3–19, 1971.

[58] S. Mahajan and G. van Ryzin. Inventory competition under dynamic consumer choice. *Operations Research* 49(5):646–657, 1999.

[59] S. Mahajan and G. van Ryzin. Supply chain coordination under horizontal competition. Working paper, Columbia University, New York, 1999.

[60] P. Majumder and H. Groenevelt. Competition in remanufacturing. *Production and Operations Management* 10(2):125–141, 2001.

[61] P. Majumder and H. Groenevelt. Procurement competition in remanufacturing. Working paper, Duke University, 2001.

[62] H. Moulin. *Game Theory for the Social Sciences*. New York University Press, New York, 1986.

[63] H. Moulin. *Cooperative Microeconomics: A Game-Theoretic Introduction*. Princeton University Press, Princeton, NJ, 1995.

[64] S. K. Mukhopadhyay and P. Kouvelis. A differential game theoretic model for duopolistic competition on design quality. *Operations Research* 45(6):886–893, 1997.

[65] A. Muller, M. Scarsini, and M. Shaked. The newsvendor game has a nonempty core. *Games and Economic Behavior* 38(1):118–126, 2002.

[66] R. B. Myerson. *Game Theory*. Harvard University Press, Cambridge, MA, 1997.

[67] M. Nagarajan and G. Sosic. Game-theoretic analysis of cooperation among supply chain agents: Review and extensions. Technical report, University of Southern California, CA, 2005.

[68] J. F. Nash. Equilibrium points in N-person games. *Proceedings of the National Academy of Sciences of the USA* 36(1):48–49, 1950.

[69] S. Netessine and N. Rudi. Centralized and competitive inventory models with demand substitution. *Operations Research* 51(2):329–335, 2003.

[70] S. Netessine and N. Rudi. Supply chain structures on the Internet and the role of marketing-operations interaction. D. Simchi-Levi, S. D. Wu, and M. Shen, eds. *Supply Chain Analysis in E-Business Era*. Springer, New York, 2004.

[71] S. Netessine and N. Rudi. Supply chain choice on the internet. *Management Science* 52(6):844–864, 2006.

[72] S. Netessine and R. Shumsky. Revenue management games: Horizontal and vertical competition. *Management Science* 51(5):813–831, 2005.

[73] S. Netessine and F. Zhang. The impact of supply-side externalities among downstream firms on supply chain efficiency. *Manufacturing and Service Operations Management* 7(1):58–73, 2005.

[74] S. Netessine, N. Rudi, and Y. Wang. Inventory competition and incentives to backorder. *IIE Transactions* 38(11):883–902, 2006.

[75] M. Parlar. Game theoretic analysis of the substitutable product inventory problem with random demands. *Naval Research Logistics* 35(3):397–409, 1988.

[76] E. L. Plambeck and T. A. Taylor. Implications of renegotiation for optimal contract flexibility and investment. Working paper, Stanford University, Stanford, CA, 2001.

[77] E. L. Plambeck and T. A. Taylor. Sell the plant? The impact of contract manufacturing on innovation, capacity, and profitability. *Management Science* 51(1):133–150, 2005.

[78] E. Porteus and S. Whang. Supply chain contracting: Non-recurring engineering charge, minimum order quantity, and boilerplate contracts. Working paper, Stanford University, Stanford, CA, 1999.

[79] J. Ren, M. Cohen, T. Ho. and C. Terwiesch. Sharing forecast information in a long-term supply chain relationship. Working paper, University of Pennsylvania, 2003.

[80] J. B. Rosen. Existence and uniqueness of equilibrium points for concave N-person games. *Econometrica* 33(3):520–533, 1965.

[81] N. Rudi, S. Kapur, and D. Pyke. A two-location inventory model with transshipment and local decision making. *Management Science* 47(12):1668–1680, 2001.

[82] R. Selten. Spieltheoretische Behandlung eines Oligopolmodells mit Nachfragetragheit. *Zeitschrift fur die gesamte Staatswissenschaft* 12:301–324, 1965.

[83] R. Selten. Reexamination of the perfectness concept for equilibrium points in extensive games. *International Journal of Game Theory* 4:25–55, 1975.

[84] L. Shapley. Stochastic games. *Proceedings of the National Academy of Sciences of the USA* 39(1):1095–1100, 1953.

[85] L. Shapley. A value for n-person game. H. W. Kuhn and A. W. Tucker, eds. *Contributions to the Theory of Games*, Vol. II. Princeton University Press, Princeton, NJ, 1953.

[86] M. Shubik. Incentives, decentralized control, the assignment of joint costs and internal pricing. *Management Science* 8(3):325–343, 1962.

[87] M. Shubik. Game theory and operations research: Some musings 50 years later. *Operations Research* 50(1):192–196, 2002.

[88] D. Simchi-Levi, S. D. Wu, and M. Shen, eds. *Handbook of Quantitative Supply Chain Analysis: Modeling in the E-Business Era*. Springer, New York, 2004.

[89] M. J. Sobel. Noncooperative stochastic games. *Annals of Mathematical Statistics* 42(6):1930–1935, 1971.

[90] S. Stidham. Pricing and capacity decisions for a service facility: Stability and multiple local optima. *Management Science* 38(8):1121–1139, 1992.

[91] H. W. Stuart, Jr. Cooperative games and business strategy. K. Chatterjee and W. F. Samuelson, eds. *Game Theory and Business Applications*. Springer, New York, 2001.

[92] H. W. Stuart, Jr. Biform analysis of inventory competition. *Manufacturing and Service Operations Management* 7(4):347–359, 2005.

[93] A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics* 5:285–308, 1955.

[94] T. A. Taylor and E. L. Plambeck. Supply chain relationships and contracts: The impact of repeated interaction on capacity investment and procurement. Working paper, Columbia University, New York, 2003.

[95] S. Tayur and W. Yang. Equilibrium analysis of a natural gas supply chain. Working paper, Carnegie Mellon University, Pittsburgh, PA, 2002.

[96] D. M. Topkis. *Supermodularity and Complementarity*. Princeton University Press, Princeton, NJ, 1998.

[97] J. Van Mieghem. Coordinating investment, production and subcontracting. *Management Science* 45(7):954–971, 1999.

[98] J. Van Mieghem and M. Dada. Price versus production postponement: Capacity and competition. *Management Science* 45(12):1631–1649, 1999.

[99] H. Varian. A model of sales. *American Economic Review* 70(4):651–659, 1980.

[100] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance* 16(1):8–37, 1961.

[101] X. Vives. *Oligopoly Pricing: Old Ideas and New Tools*. MIT Press, Cambridge, MA, 1999.

[102] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, NJ, 1944.

[103] H. von Stackelberg. *Marktform und Gleichgewicht*. Julius Springer, Vienna, Austria, 1934.

[104] Y. Wang and Y. Gerchak. Capacity games in assembly systems with uncertain demand. *Manufacturing and Service Operations Management* 5(3):252–267, 2003.

[105] Q. Wang and M. Parlar. Static game theory models and their applications in management science. *European Journal of Operational Research* 42(1):1–21, 1989.

[106] Q. Wang and M. Parlar. A three-person game theory model arising in stochastic inventory control theory. *European Journal of Operational Research* 76(1):83–97, 1994.

TUTORIALS IN
OPERATIONS RESEARCH
INFORMS 2006

 | ISBN 13 978-1-877640-20-9
DOI 10.1287/educ.1063.0025

Planning for Disruptions in Supply Chain Networks

Lawrence V. Snyder
Department of Industrial and Systems Engineering, Lehigh University, Mohler Lab, 200 West Packer Avenue, Bethlehem, Pennsylvania 18013, larry.snyder@lehigh.edu

Maria P. Scaparra
Kent Business School, University of Kent, Canterbury, CT2 7PE, England, m.p.scaparra@kent.ac.uk

Mark S. Daskin
Department of Industrial Engineering and Management Sciences, Northwestern University, 2145 Sheridan Road, Evanston, Illinois 60208, m-daskin@northwestern.edu

Richard L. Church
Department of Geography, University of California, Santa Barbara, California 593106-4060, church@geog.ucsb.edu

Abstract Recent events have highlighted the need for planners to consider the risk of disruptions when designing supply chain networks. Supply chain disruptions have a number of causes and may take a number of forms. Once a disruption occurs, there is very little recourse regarding supply chain infrastructure because these strategic decisions cannot be changed quickly. Therefore, it is critical to account for disruptions during the design of supply chain networks so that they perform well even after a disruption. Indeed, these systems can often be made substantially more reliable with only small additional investments in infrastructure. Planners have a range of options available to them in designing resilient supply chain networks, and their choice of approaches will depend on the financial resources available, the decision maker's risk preference, the type of network under consideration, and other factors. In this tutorial, we present a broad range of models for designing supply chains resilient to disruptions. We first categorize these models by the status of the existing network: A network may be designed from scratch, or an existing network may be modified to prevent disruptions at some facilities. We next divide each category based on the underlying optimization model (facility location or network design) and the risk measure (expected cost or worst-case cost).

Keywords facility location; network design; disruptions

1. Introduction

1.1. Motivation

Every supply chain faces disruptions of various sorts. Recent examples of major disruptions are easy to bring to mind: Hurricanes Katrina and Rita in 2005 on the U.S. Gulf Coast crippled the nation's oil refining capacity (Mouawad [68]), destroyed large inventories of coffee and lumber (Barrionuevo and Deutsch [3], Reuters [74]), and forced the rerouting of bananas and other fresh produce (Barrionuevo and Deutsch [3]). A strike at two General Motors parts plants in 1998 led to the shutdowns of 26 assembly plants, which ultimately

resulted in a production loss of over 500,000 vehicles and an $809 million quarterly loss for the company (Brack [13], Simison [88, 89]). An eight-minute fire at a Philips semiconductor plant in 2001 brought one customer, Ericsson, to a virtual standstill while another, Nokia, weathered the disruption (Latour [58]). Moreover, smaller-scale disruptions occur much more frequently. For example, Wal-Mart's Emergency Operations Center receives a call virtually every day from a store or other facility with some sort of crisis (Leonard [60]).

There is evidence that superior contingency planning can significantly mitigate the effect of a disruption. For example, Home Depot's policy of planning for various types of disruptions based on geography helped it get 23 of its 33 stores within Katrina's impact zone open after one day and 29 after one week (Fox [37]), and Wal-Mart's stock prepositioning helped make it a model for post-hurricane recovery (Leonard [60]). Similarly, Nokia weathered the 2001 Phillips fire through superior planning and quick response, ultimately allowing it to capture a substantial portion of Ericsson's market share (Latour [58]).

Recent books and articles in the business and popular press have pointed out the vulnerability of today's supply chains to disruptions and the need for a systematic analysis of supply chain vulnerability, security, and resiliency (Elkins et al. [35], Jüttner et al. [52], Lynn [63], Rice and Caniato [76], Sheffi [84]). One common theme among these references is that the tightly optimized, just-in-time, lean supply chain practices championed by practitioners and OR researchers in recent decades increase the vulnerability of these systems. Many have argued that supply chains should have more redundancy or slack to provide a buffer against various sorts of uncertainty. Nevertheless, companies have historically been reluctant to invest much in additional supply chain infrastructure or inventory, despite the large payoff that such investments can have if a disruption occurs.

We argue that decision makers should take supply uncertainty (of which disruptions are one variety) into account during all phases of supply chain planning, just as they account for demand uncertainty. This is most critical during strategic planning because these decisions cannot easily be modified. When a disruption strikes, there is very little recourse for strategic decisions like facility location and network design. (In contrast, firms can often adjust inventory levels, routing plans, production schedules, and other tactical and operational decisions in real time in response to unexpected events.)

It is easy to view supply uncertainty and demand uncertainty as two sides of the same coin. For example, a toy manufacturer may view stockouts of a hot new toy as a result of demand uncertainty, but to a toy store, the stockouts look like a supply-uncertainty issue. Many techniques that firms use to mitigate demand uncertainty—safety stock, supplier redundancy, forecast refinements—also apply in the case of supply uncertainty. However, it is dangerous to assume that supply uncertainty is a special case of demand uncertainty or that it can be ignored by decision makers, because much of the conventional wisdom gained from studying demand uncertainty does not hold under supply uncertainty. For example, under demand uncertainty, it may be optimal for a firm to operate fewer distribution centers (DCs) because of the risk-pooling effect and economies of scale in ordering (Daskin et al. [27]), while under supply uncertainty, it may be optimal to operate more, smaller DCs so that a disruption to one of them has lesser impact. Snyder and Shen [95] discuss this and other differences between the two forms of uncertainty.

In this tutorial, we discuss models for designing supply chain networks that are resilient to disruptions. The objective is to design the supply chain infrastructure so that it operates efficiently (i.e., at low cost) both normally and when a disruption occurs. We discuss models for facility location and network design. Additionally, we analyze fortification models that can be used to improve the reliability of infrastructure systems already in place and for which a complete reconfiguration would be cost prohibitive. The objective of fortification models is to identify optimal strategies for allocating limited resources among possible mitigation investments.

1.2. Taxonomy and Tutorial Outline

We classify models for reliable supply chain design along three axes.

(1) *Design vs. fortification.* Is the model intended to create a reliable network assuming that no network is currently in place, or to fortify an existing network to make it more reliable?

(2) *Underlying model.* Reliability models generally have some classical model as their foundation. In this tutorial, we consider models based on facility location and network design models.

(3) *Risk measure.* As in the case of demand uncertainty, models with supply uncertainty need some measure for evaluating risk. Examples include expected cost and minimax cost.

This tutorial is structured according to this taxonomy. Section 3 discusses design models, while §4 discusses fortification models, with subsections in each to divide the models according to the remaining two axes. These sections are preceded by a review of the related literature in §2 and followed by conclusions in §5.

2. Literature Review

We discuss the literature that is directly related to reliable supply chain network design throughout this tutorial. In this section, we briefly discuss several streams of research that are indirectly related. For more detailed reviews of facility location models under uncertainty, the reader is referred to Daskin et al. [29], Owen and Daskin [70], and Snyder [90]. See Daskin [26] or Drezner [33] for a textbook treatment of facility location theory. An excellent overview of stochastic programming theory in general is provided in Higle [45].

2.1. Network Reliability Theory

The concept of supply chain reliability is related to network reliability theory (Colbourn [22], Shier [86], Shooman [87]), which is concerned with calculating or maximizing the probability that a graph remains connected after random failures due to congestion, disruptions, or blockages. Typically, this literature considers disruptions to the links of a network, but some papers consider node failures (Eiselt et al. [34]), and in some cases the two are equivalent. Given the difficulty in computing the reliability of a given network, the goal is often to find the minimum-cost network with some desirable property like two-connectivity (Monma [66], Monma and Shalcross [67]), k-connectivity (Bienstock et al. [11], Grötschel et al. [41]), or special ring structures (Fortz and Labbe [36]). The key difference between network reliability models and the models we discuss in this tutorial is that network reliability models are primarily concerned with connectivity; they consider the cost of constructing the network but not the cost that results from a disruption, whereas our models consider both types of costs and generally assume connectivity after a disruption.

2.2. Vector-Assignment Problems

Weaver and Church [104] introduce the *vector-assignment P-median problem* (VAPMP), in which each customer is assigned to several open facilities according to an exogenously determined frequency. For example, a customer might receive 75% of its demand from its nearest facility, 20% from its second nearest, and 5% from its third nearest. This is similar to the assignment strategy used in many of the models below, but in our models the percentages are determined endogenously based on disruptions rather than given as inputs to the model. A vector-assignment model based on the uncapacitated fixed-charge location problem (UFLP) is presented by Pirkul [73].

2.3. Multiple, Excess, and Backup Coverage Models

The maximum covering problem (Church and ReVelle [17]) locates a fixed number of facilities to maximize the demands located within some radius of an open facility. It implicitly

assumes that the facilities (e.g., fire stations, ambulances) are always available. Several subsequent papers have considered the congestion at facilities when multiple calls are received at the same time. The maximum expected covering location model (MEXCLM) (Daskin [24, 25]) maximizes the expected coverage given a constant, systemwide probability that a server is busy at any given time. The constant-busy-probability assumption is relaxed in the maximum availability location problem (MALP) (ReVelle and Hogan [75]). A related stream of research explicitly considers the queueing process at the locations; these "hypercube" models are interesting as descriptive models but are generally too complex to embed into an optimization framework (Berman et al. [10], Larson [56, 57]). See Berman and Krass [6] and Daskin et al. [28] for a review of expected and backup coverage models. The primary differences between these models and the models we discuss in this tutorial are (1) the objective function (coverage versus cost), and (2) the reason for a server's unavailability (congestion versus disruptions).

2.4. Inventory Models with Supply Disruptions

There is a stream of research in the inventory literature that considers supply disruptions in the context of classical inventory models, such as the EOQ (Parlar and Berkin [72], Berk and Arreola-Risa [5], Snyder [91]), (Q, R) (Gupta [42], Parlar [71], Mohebbi [64, 65]), and (s, S) (Arreola-Risa and DeCroix [1]) models. More recent models examine a range of strategies for mitigating disruptions, including dual sourcing (Tomlin [100]), demand management (Tomlin [99]), supplier reliability forecasting (Tomlin [98], Tomlin and Snyder [101]), and product-mix flexibility (Tomlin and Wang [102]). Few models consider disruptions in multiechelon supply chain or inventory systems; exceptions include Kim et al. [53], Hopp et al. [47], and Snyder and Shen [95].

2.5. Process Flexibility

At least five strategies can be employed in the face of uncertain demands: Expanding capacity, holding reserve inventory, improving demand forecasts, introducing product commonality to delay the need for specialization, and adding flexibility to production plants. A complete review of each strategy is beyond the scope of this tutorial. Many of these strategies are fairly straightforward. Process flexibility, on the other hand, warrants a brief discussion. Jordan and Graves [51] compare the expected lost sales that result from using a set of fully flexible plants, in which each plant could produce each product, to a configuration in which each plant produces only two products and the products are chained in such a way that plant A produces products 1 and 2, plant B produces products 2 and 3, and so on, with the last plant producing the final product as well as product 1. They refer to this latter configuration as a 1-chain. They find that a 1-chain provides nearly all of the benefits of total flexibility when measured by the expected number of lost sales. Based on this, they recommend that flexibility be added to create fewer, longer chains of products and plants. Bish et al. [12] study capacity allocation schemes for such chains (e.g., allocate capacity to the nearest demands, to the highest-margin demands, or to a plant's primary product). They find that if the capacity is either very small or very large relative to the expected demand, the gains from managing flexible capacity are outweighed by the need for additional component inventory at the plants and the costs of order variability at suppliers. They then provide guidelines for the use of one allocation policy relative to others based on the costs of component inventory, component lead times, and profit margins. Graves and Tomlin [40] extend the Jordan and Graves results to multistage systems. They contrast configuration loss with configuration inefficiency. The former measures the difference between the shortfall with total flexibility and the shortfall with a particular configuration of flexible plants. The configuration inefficiency measures the effect of the interaction between stages in causing the shortfall for a particular configuration. They show that this, in turn, is caused by two phenomena: Floating bottlenecks and stage-spanning bottlenecks. Stage-spanning bottlenecks

can arise even if demand is deterministic, as a result of misallocations of capacity across the various stages of the supply chain. Beach et al. [4] and de Toni and Tonchia [30] provide more detailed reviews of the manufacturing flexibility literature.

2.6. Location of Protection Devices

A number of papers in the location literature have addressed the problem of finding the optimal location of protection devices to reduce the impact of possible disruptions to infrastructure systems. For example, Carr et al. [16] present a model for optimizing the placement of sensors in water supply networks to detect maliciously injected contaminants. James and Salhi [49] investigate the problem of placing protection devices in electrical supply networks to reduce the amount of outage time. Flow-interception models (Berman et al. [7]) have also been used to locate protection facilities. For example, Hodgson et al. [46] and Gendreau et al. [39] use flow-interception models to locate inspection stations so as to maximize hazard avoidance and risk reduction in transportation networks. The protection models discussed in this chapter differ from those models in that they do not seek the optimal placement of physical protection devices or facilities. Rather, they aim at identifying the most critical system components to harden or protect with limited protection resources (for example, through structural retrofit, fire safety, increased surveillance, vehicle barriers, and monitoring systems).

3. Design Models

3.1. Introduction

In this section, we discuss *design models* for reliable facility location and network design. These models, like most facility location models, assume that no facilities currently exist; they aim to choose a set of facility locations that perform well even if disruptions occur. It is also straightforward to modify these models to account for facilities that may already exist (e.g., by setting the fixed cost of those facilities to zero or adding a constraint that requires them to be open). In contrast, the *fortification models* discussed in §4 assume that all facility sites have been chosen and attempt to decide which facilities to fortify (protect against disruptions). One could conceivably formulate an integrated design/fortification model whose objective would be to locate facilities and identify a subset of those facilities to fortify against attacks. Formulation of such a model is a relatively straightforward extension of the models we present below, though its solution would be considerably more difficult because it would result in (at least) a tri-level optimization problem.

Most models for both classical and reliable facility location are design models, because as "fortification" is a relatively new concept in the facility location literature. In the subsections that follow, we introduce several design models, classified first according to the underlying model (facility location or network design) and then according to risk measure (expected or worst-case cost).

3.2. Facility Location Models

3.2.1. Expected Cost Models. In this section, we define the reliability fixed-charge location problem (RFLP) (Snyder and Daskin [92]), which is based on the classical uncapacitated fixed-charge location problem (UFLP) (Balinski [2]). There is a fixed set I of customer locations and a set J of potential facility locations. Each customer $i \in I$ has an annual demand of h_i units, and each unit shipped from facility $j \in J$ to customer $i \in I$ incurs a transportation cost of d_{ij}. (We will occasionally refer to d_{ij} as the "distance" between j and i, and use this notion to refer to "closer" or "farther" facilities.) Each facility site has an annual fixed cost f_j that is incurred if the facility is opened. Any open facility may serve any customer (that is, there are no connectivity restrictions), and facilities have unlimited capacity. There is a single product.

Each open facility may fail (be disrupted) with a fixed probability q. (Note that the failure probability q is the same at every facility. This assumption allows a compact description of the expected transportation cost. Below, we relax this assumption and instead formulate a scenario-based model that requires more decision variables but is more flexible.) Failures are independent, and multiple facilities may fail simultaneously. When a facility fails, it cannot provide any product, and the customers assigned to it must be reassigned to a nondisrupted facility.

If customer i is not served by any facility, the firm incurs a penalty cost of θ_i per unit of demand. This penalty may represent a lost-sales cost or the cost of finding an alternate source for the product. It is incurred if all open facilities have failed, or if it is too expensive to serve a customer from its nearest functional facility. To model this, we augment the facility set J to include a dummy "emergency facility," called u, that has no fixed cost ($f_u = 0$) and never fails. The transportation cost from u to i is $d_{iu} \equiv \theta_i$. Assigning a customer to the emergency facility is equivalent to not assigning it at all.

The RFLP uses two sets of decision variables:

$$X_j = \begin{cases} 1, & \text{if facility } j \text{ is opened,} \\ 0, & \text{otherwise,} \end{cases}$$

$$Y_{ijr} = \begin{cases} 1, & \text{if customer } i \text{ is assigned to facility } j \text{ at level } r, \\ 0, & \text{otherwise.} \end{cases}$$

A "level-r" assignment is one for which there are r closer open facilities. For example, suppose that the three closest open facilities to customer i are facilities 2, 5, and 8, in that order. Then facility 2 is i's level-0 facility, 5 is its level-1 facility, and 8 is its level-2 facility. Level-0 assignments are to "primary" facilities that serve the customer under normal circumstances, while level-r assignments ($r > 0$) are to "backup" facilities that serve it if all closer facilities have failed. A customer must be assigned to *some* facility at each level r unless it is assigned to the emergency facility at some level $s \le r$. Because we do not know in advance how many facilities will be open, we extend the index r from 0 through $|J| - 1$, but Y_{ijr} will equal 0 for r greater than or equal to the number of open facilities.

The objective of the RFLP is to choose facility locations and customer assignments to minimize the fixed cost plus the expected transportation cost and lost-sales penalty. We formulate it as an integer programming problem as follows.

$$\text{(RFLP)} \quad \text{minimize} \quad \sum_{j\in J} f_j X_j + \sum_{i\in I} \sum_{r=0}^{|J|-1} \left[\sum_{j\in J\setminus\{u\}} h_i d_{ij} q^r (1-q) Y_{ijr} + h_i d_{iu} q^r Y_{iur} \right] \tag{1}$$

subject to

$$\sum_{j\in J} Y_{ijr} + \sum_{s=0}^{r-1} Y_{ius} = 1 \quad \forall\, i \in I,\ r = 0, \ldots, |J| - 1 \tag{2}$$

$$Y_{ijr} \le X_j \quad \forall\, i \in I,\ j \in J,\ r = 0, \ldots, |J| - 1 \tag{3}$$

$$\sum_{r=0}^{|J|-1} Y_{ijr} \le 1 \quad \forall\, i \in I,\ j \in J \tag{4}$$

$$X_j \in \{0, 1\} \quad \forall\, j \in J \tag{5}$$

$$Y_{ijr} \in \{0, 1\} \quad \forall\, i \in I,\ j \in J,\ r = 0, \ldots, |J| - 1 \tag{6}$$

The objective function (1) minimizes the sum of the fixed cost and the expected transportation and lost-sales costs. The second term reflects the fact that if customer i is assigned to facility j at level r, it will actually be served by j if all r closer facilities have failed (which

happens with probability q^r) and if j itself has not failed (which happens with probability $1-q$). Note that we can compute this expected cost knowing only the *number* of facilities that are closer to i than j, but not which facilities those are. This is a result of our assumption that every facility has the same failure probability. If, instead, customer i is assigned to the emergency facility at level r, then it incurs the lost-sales cost $d_{iu} \equiv \theta_i$ if its r closest facilities have failed (which happens with probability q^r).

Constraints (2) require each customer i to be assigned to some facility at each level r, unless i has been assigned to the emergency facility at level $s < r$. Constraints (3) prevent an assignment to a facility that has not been opened, and constraints (4) prohibit a customer from being assigned to the same facility at more than one level. Constraints (5) and (6) require the decision variables to be binary. However, constraints (6) can be relaxed to nonnegativity constraints because single sourcing is optimal in this problem, as it is in the UFLP.

Note that we do not explicitly enforce the definition of "level-r assignment" in this formulation; that is, we do not require $Y_{ijr} = 1$ only if there are exactly r closer open facilities. Nevertheless, in any optimal solution, this definition will be satisfied because it is optimal to assign customers to facilities by levels in increasing order of distance. This is true because the objective function weights decrease for larger values of r, so it is advantageous to use facilities with smaller d_{ij} at smaller assignment levels. A slight variation of this result is proven rigorously by Snyder and Daskin [92].

Snyder and Daskin [92] present a slightly more general version of this model in which some of the facilities may be designated as "nonfailable." If a customer is assigned to a nonfailable facility at level r, it does not need to be assigned at any higher level. In addition, Snyder and Daskin [92] consider a multiobjective model that minimizes the weighted sum of two objectives, one of which corresponds to the UFLP cost (fixed cost plus level-0 transportation costs) while the other represents the expected transportation cost (accounting for failures). By varying the weights on the objectives, Snyder and Daskin [92] generate a trade-off curve and use this to demonstrate that the RFLP can produce solutions that are much more reliable than the classical UFLP solution but only slightly more expensive by the UFLP objective. This suggests that reliability can be "bought" relatively cheaply. Finally, Snyder and Daskin [92] also consider a related model that is based on the P-median problem (Hakimi [43, 44]) rather than the UFLP. They solve all models using Lagrangian relaxation.

In general, the optimal solution to the RFLP uses more facilities than that of the UFLP. This tendency toward diversification occurs so that any given disruption affects a smaller portion of the system. It may be viewed as a sort of "risk-diversification effect" in which it is advantageous to spread the risk of supply uncertainty across multiple facilities (encouraging decentralization). This is in contrast to the classical risk-pooling effect, which encourages centralization to pool the risk of demand uncertainty (Snyder and Shen [95]).

Berman et al. [8] consider a model similar to (RFLP), based on the P-median problem rather than the UFLP. They allow different facilities to have different failure probabilities, but the resulting model is highly nonlinear and, in general, must be solved heuristically. They prove that the Hakimi property applies if colocation is allowed. (The Hakimi property says that optimal locations exist at the nodes of a network, even if facilities are allowed on the links.) Berman et al. [9] present a variant of this model in which customers do not know which facilities are disrupted before visiting them and must traverse a path from one facility to the next until an operational facility is found. For example, a customer might walk to the nearest ATM, find it out of order, and then walk to the ATM that is nearest to the current location. They investigate the spatial characteristics of the optimal solution and discuss the value of reliability information.

An earlier attempt at addressing reliability issues in P-median problems is discussed by Drezner [32], who examines the problem of locating P unreliable facilities in the plane so as to minimize expected travel distances between customers and facilities. As in the RFLP,

the unreliable P-median problem in Drezner [32] is defined by introducing a probability that a facility becomes inactive but does not require the failures to be independent events. The problem is solved through a heuristic procedure. A more sophisticated method to solve the unreliable P-median problem was subsequently proposed in Lee [59]. Drezner [32] also presents the unreliable (P, Q)-center problem where P facilities must be located while taking into account that Q of them may become unavailable simultaneously. The objective is to minimize the maximal distance between demand points and their closest facilities.

The formulation given above for (RFLP) captures the expected transportation cost without using explicit scenarios to describe the uncertain events (disruptions). An alternate approach is to model the problem as a two-stage stochastic programming problem in which the location decisions are first-stage decisions and the assignment decisions are made in the second stage, after the random disruptions have occurred. This approach can result in a much larger IP model because $2^{|J|}$ possible failure scenarios exist, and each requires its own assignment variables. That is, in the formulation above we have $|J|$ Y variables for each i, j (indexed Y_{ijr}, $r = 0, \ldots, |J| - 1$), while in the scenario-based formulation we have $2^{|J|}$ variables for each i, j. However, formulations built using this approach can be solved using standard stochastic programming methods. They can also be adapted more readily to handle side constraints and other variations.

For example, suppose facility j can serve at most b_j units of demand at any given time. These capacity constraints must be satisfied both by "primary" assignments and by reassignments that occur after disruptions. Let S be the set of failure scenarios such that $a_{js} = 1$ if facility j fails in scenario s, and let q_s be the probability that scenario s occurs. Finally, let Y_{ijs} equal 1 if customer i is assigned to facility j in scenario s and 0 otherwise. The capacitated RFLP can be formulated using the scenario-based approach as follows.

$$\text{(CRFLP)} \quad \text{minimize} \quad \sum_{j \in J} f_j X_j + \sum_{s \in S} q_s \sum_{i \in I} \sum_{j \in J} h_i d_{ij} Y_{ijs} \tag{7}$$

subject to

$$\sum_{j \in J} Y_{ijs} = 1 \quad \forall i \in I, s \in S \tag{8}$$

$$Y_{ijs} \leq X_j \quad \forall i \in I,\ j \in J,\ s \in S \tag{9}$$

$$\sum_{i \in I} h_i Y_{ijs} \leq (1 - a_{js}) b_j \quad \forall j \in J,\ s \in S \tag{10}$$

$$X_j \in \{0, 1\} \quad \forall j \in J \tag{11}$$

$$Y_{ijs} \in \{0, 1\} \quad \forall i \in I,\ j \in J,\ s \in S \tag{12}$$

Note that the set J in this formulation still includes the emergency facility u. The objective function (7) computes the sum of the fixed cost plus the expected transportation cost, taken across all scenarios. Constraints (8) require every customer to be assigned to some facility (possibly u) in every scenario, and constraints (9) require this facility to be opened. Constraints (10) prevent the total demand assigned to facility j in scenario s from exceeding j's capacity and prevent any demand from being assigned if the facility has failed in scenario s. Constraints (11) and (12) are integrality constraints. Integrality can be relaxed to nonnegativity for the Y variables if single-sourcing is not required. (Single-sourcing is no longer optimal because of the capacity constraints.)

(CRFLP) can be modified easily without destroying its structure, in a way that (RFLP) cannot. For example, if the capacity during a disruption is reduced but not eliminated, we can simply redefine a_{js} to be the proportion of the total capacity that is affected by the disruption. We can also easily allow the demands and transportation costs to be scenario dependent.

The disadvantage, of course, is that the number of scenarios grows exponentially with $|J|$. If $|J|$ is reasonably large, enumerating all of the scenarios is impractical. In this case, one generally must use sampling techniques such as sample average approximation (SAA) (Kleywegt et al. [54], Linderoth et al. [62], Shapiro and Homem-de-Mello [83]), in which the optimization problem is solved using a subset of the scenarios sampled using Monte Carlo simulation. By solving a series of such problems, one can develop bounds on the optimal objective value and the objective value of a given solution. Ülker and Snyder [103] present a method for solving (CRFLP) that uses Lagrangian relaxation embedded in an SAA scheme.

An ongoing research project has focused on extending the models discussed in this section to account for inventory costs when making facility location decisions. Jeon et al. [50] consider facility failures in a location-inventory context that is similar to the models proposed recently by Daskin et al. [27] and Shen et al. [85], which account for the cost of cycle and safety stock. The optimal number of facilities in the models by Daskin et al. [27] and Shen et al. [85] is smaller than those in the UFLP due to economies of scale in ordering and the risk-pooling effect. Conversely, the optimal number of facilities is larger in the RFLP than in the UFLP to reduce the impact of any single disruption. The location-inventory model with disruptions proposed by Jeon et al. [50] finds a balance between these two competing tendencies.

3.2.2. Worst-Case Cost Models. Models that minimize the expected cost, as in §3.2.1, take a risk-neutral approach to decision making under uncertainty. Risk-averse decision makers may be more inclined to minimize the worst-case cost, taken across all scenarios. Of course, in this context, it does not make sense to consider all possible scenarios, because otherwise the worst-case scenario is always the one in which *all* facilities fail. Instead, we might consider all scenarios in which, say, at most three facilities fail, or all scenarios with probability at least 0.01, or some other set of scenarios identified by managers as worth planning against. In general, the number of scenarios in such a problem is smaller than in the expected-cost problem because scenarios that are clearly less costly than other scenarios can be omitted from consideration. For example, if we wish to consider scenarios in which at most three facilities fail, we can ignore scenarios in which two or fewer fail.

To formulate the minimax-cost RFLP, we introduce a single additional decision variable U, which equals the maximum cost.

$$\text{(MMRFLP)} \quad \text{minimize} \quad U \tag{13}$$

subject to

$$\sum_{j\in J} f_j X_j + \sum_{i\in I}\sum_{j\in J} h_i d_{ij} Y_{ijs} \le U \quad \forall\, s \in S \tag{14}$$

$$\sum_{j\in J} Y_{ijs} = 1 \quad \forall\, i \in I,\ s \in S \tag{15}$$

$$Y_{ijs} \le (1 - a_{js}) X_j \quad \forall\, i \in I,\ j \in J,\ s \in S \tag{16}$$

$$X_j \in \{0,1\} \quad \forall\, j \in J \tag{17}$$

$$Y_{ijs} \in \{0,1\} \quad \forall\, i \in I,\ j \in J,\ s \in S \tag{18}$$

In this formulation, we omit the capacity constraints (10), but they can be included without difficulty. Unfortunately, minimax models tend to be much more difficult to solve exactly, either with general-purpose IP solvers or with customized algorithms. This is true for classical problems as well as for (MMRFLP).

The *regret* of a solution under a given scenario is the relative or absolute difference between the cost of the solution under that scenario and the optimal cost under that scenario. One can modify (MMRFLP) easily to minimize the maximum regret across all scenarios by replacing the right side of (14) with $U + z_s$ (for absolute regret) or $z_s(1 + U)$ (for relative regret).

Here, z_s is the optimal cost in scenario s, which must be determined exogenously for each scenario and provided as an input to the model.

Minimax-regret problems may require more scenarios than their minimax-cost counterparts because it is not obvious a priori which scenarios will produce the maximum regret. On the other hand, they tend to result in a less pessimistic solution than minimax-cost models do. Snyder and Daskin [94] discuss minimax-cost and minimax-regret models in further detail.

One common objection to minimax models is that they are overly conservative because the resulting solution plans against a single scenario, which may be unlikely even if it is disastrous. In contrast, expected-cost models like the CRFLP produce solutions that perform well in the long run but may perform poorly in some scenarios. Snyder and Daskin [94] introduce a model that avoids both problems by minimizing the expected cost (7) subject to a constraint on the maximum cost that can occur in any scenario (in effect, treating U as a constant in (14)). An optimal solution to this model is guaranteed to perform well in the long run (due to the objective function) but is also guaranteed not to be disastrous in any given scenario. This approach is closely related to the concept of p-robustness in robust optimization problems (Kouvelis and Yu [55], Snyder and Daskin [93]). One computational disadvantage is that, unlike the other models we have discussed, it can be difficult (even NP-complete) to find a *feasible* solution or to determine whether a given instance is feasible. See Snyder and Daskin [94] for more details on this model and for a discussion of reliable facility location under a variety of other risk measures.

Church et al. [20] use a somewhat different approach to model worst-case cost design problems, the rationale being that the assumption of independent facility failures underlying the previous models does not hold in all application settings. This is particularly true when modeling intentional disruptions. As an example, a union or a terrorist could decide to strike those facilities in which the greatest combined harm (as measured by increased costs, disrupted service, etc.) is achieved. To design supply systems able to withstand intentional harms by intelligent perpetrators, Church et al. [20] propose the resilient P-median problem. This model identifies the best location of P facilities so that the system works as well as possible (in terms of weighted distances) in the event of a maximally disruptive strike. The model is formulated as a bilevel optimization model, in which the upper-level problem of optimally locating P facilities embeds a lower-level optimization problem used to generate the weighted distance after a worst-case loss of R of these located P facilities. This bilevel programming approach has been widely used to assess worst-case scenarios and identify critical components in existing systems and will be discussed in more depth in §4.2.2. Church et al. [20] demonstrate that optimal P-median configurations can be rendered very inefficient in terms of worst-case loss, even for small values of R. They also demonstrate that resilient design configurations can be near optimal in efficiency as compared to the optimal P-median configurations, but at the same time, maintain high levels of efficiency after worst-case loss. A form of the resilient design problem has also been developed for a coverage-type service system (O'Hanley and Church [69]). The resilient coverage model finds the optimal location of a set of facilities to maximize a combination of initial demand coverage and the minimum coverage level following the loss of one or more facilities. There are several approaches that one can employ to solve this problem, including the successive use of super-valid inequalities (O'Hanley and Church [69]), reformulation into a single-level optimization problem when $R = 1$ or $R = 2$ (Church et al. [20]), or by developing a special search tree. Research is underway to model resilient design for capacitated problems.

3.3. Network Design Models

We now turn our attention from reliability models based on facility location problems to those based on network design models. We have a general network $G = (V, A)$. Each node $i \in V$ serves as either a source, sink, or transshipment node. Source nodes are analogous

to facilities in §3.2 while sink nodes are analogous to customers. The primary difference between network design models and facility location ones is the presence of transshipment nodes. Product originates at the source nodes and is sent through the network to the sink nodes via transshipment nodes.

Like the facilities in §3.2, the nonsink nodes in these models can fail randomly. The objective is to make open/close decisions on the nonsink nodes (first-stage variables) and determine the flows on the arcs in each scenario (second-stage variables) to minimize the expected or worst-case cost. (Many classical network design problems involve open/close decisions on arcs, but the two are equivalent through a suitable transformation.)

3.3.1. Expected Cost. Each node $j \in V$ has a *supply* b_j. For source nodes, b_j represents the available supply and $b_j > 0$; for sink nodes, b_j represents the (negative of the) demand and $b_j < 0$; for transshipment nodes, $b_j = 0$. There is a fixed cost f_j to open each nonsink node. Each arc (i, j) has a cost of d_{ij} for each unit of flow transported on it, and each nonsink node j has a capacity k_j. The node capacities can be seen as production limitations for the supply nodes and processing resource restrictions for the transhipment nodes.

As in §3.2.1, we let S be the set of scenarios, and $a_{js} = 1$ if node j fails in scenario s. Scenario s occurs with probability q_s. To ensure feasibility in each scenario, we augment V by adding a dummy source node u that makes up any supply shortfall caused by disruptions and a dummy sink node v that absorbs any excess supply. There is an arc from u to each (nondummy) sink node; the per-unit cost of this arc is equal to the lost-sales cost for that sink node (analogous to θ_i in §3.2.1). Similarly, there is an arc from each (nondummy) source node to v whose cost equals 0. The dummy source node and the dummy sink node have infinite supply and demand, respectively.

Let $V_0 \subseteq V$ be the set of supply and transhipment nodes, i.e., $V_0 = \{j \in V \mid b_j \geq 0\}$. We define two sets of decision variables. $X_j = 1$ if node i is opened and 0 otherwise, for $j \in V_0$, and Y_{ijs} is the amount of flow sent on arc $(i, j) \in A$ in scenario $s \in S$. Note that the set A represents the augmented set of arcs, including the arcs outbound from the dummy source node and the arcs inbound to the dummy sink node. With this notation, the reliable network design model (RNDP) is formulated as follows.

$$\text{(RNDP)} \quad \text{minimize} \quad \sum_{j \in V_0} f_j X_j + \sum_{s \in S} q_s \sum_{(i,j) \in A} d_{ij} Y_{ijs} \tag{19}$$

subject to

$$\sum_{(j,i) \in A} Y_{jis} - \sum_{(i,j) \in A} Y_{ijs} = b_j \quad \forall j \in V \backslash \{u, v\},\ s \in S \tag{20}$$

$$\sum_{(j,i) \in A} Y_{jis} \leq (1 - a_{js}) k_j X_j \quad \forall j \in V_0,\ s \in S \tag{21}$$

$$X_j \in \{0, 1\} \quad \forall j \in V_0 \tag{22}$$

$$Y_{ijs} \geq 0 \quad \forall (i, j) \in A,\ s \in S \tag{23}$$

The objective function computes the fixed cost and expected flow costs. Constraints (20) are the flow-balance constraints for the nondummy nodes; they require the net flow for node j (flow out minus flow in) to equal the node's deficit b_j in each scenario. Constraints (21) enforce the node capacities and prevent any flow emanating from a node j that has not been opened ($X_j = 0$) or has failed ($a_{js} = 1$). Taken together with (20), these constraints are sufficient to ensure that flow is also prevented *into* nodes that are not opened or have failed. Constraints (22) and (23) are integrality and nonnegativity constraints, respectively. Note that in model (19)–(23), no flow restrictions are necessary for the two dummy nodes. The minimization nature of the objective function guarantees that the demand at each sink node is supplied from regular source nodes whenever this is possible. Only if the node disruption is such to prevent some demand node i from being fully supplied will there be a positive

flow on the link (u, i) at the cost $d_{ui} = \theta_i$. Similarly, only excess supply that cannot reach a sink node will be routed to the dummy sink.

This formulation is similar to the model introduced by Santoso et al. [78]. Their model is intended for network design under demand uncertainty, while ours considers supply uncertainty, though the two approaches are quite similar. To avoid enumerating all possible scenarios, Santoso et al. [78] use SAA. A similar approach is called for to solve (RNDP) because, as in the scenario-based models in §3.2.1, if each node can fail independently, we have $2^{|V_0|}$ scenarios.

A scenario-based model for the design of failure-prone multicommodity networks is discussed in Garg and Smith [38]. However, the model in Garg and Smith [38] does not consider the expected costs of routing the commodities through the network. Rather, it determines the minimum-cost set of arcs to be constructed so that the resulting network continues to support a multicommodity flow under any of a given set of failure scenarios. Only a restricted set of failure scenarios is considered, in which each scenario consists of the concurrent failure of multiple arcs. Garg and Smith [38] also discuss several algorithmic implementations of Benders decomposition to solve this problem efficiently.

3.3.2. Worst-Case Cost. One can modify (RNDP) to minimize the worst-case cost rather than the expected cost in a manner analogous to the approach taken in §3.2.2.

$$\text{minimize} \quad U \tag{24}$$

$$\text{subject to} \quad \sum_{i \in V_0} f_i X_i + \sum_{(i,j) \in A} d_{ij} Y_{ijs} \leq U \quad \forall\, s \in S \tag{25}$$

$$(20)\text{–}(23)$$

Similarly, one could minimize the expected cost subject to a constraint on the cost in any scenario, as proposed above. Bundschuh et al. [15] take a similar approach in a supply chain network design model (with open/close decisions on arcs). They assume that suppliers can fail randomly. They consider two performance measures, which they call *reliability* and *robustness*. The reliability of the system is the probability that all suppliers are operable, while robustness refers to the ability of the supply chain to maintain a given level of output after a failure. The latter measure is perhaps a more reasonable goal because adding new suppliers increases the probability that one or more will fail and, hence, *decreases* the system's "reliability." They present models for minimizing the fixed and (nonfailure) transportation costs subject to constraints on reliability, robustness, or both. Their computational results support the claim made by Snyder and Daskin [92, 94] and others that large improvements in reliability can often be attained with small increases in cost.

4. Fortification Models

4.1. Introduction

Computational studies of the models discussed in the previous sections demonstrate that the impact of facility disruptions can be mitigated by the initial design of a system. However, redesigning an entire system is not always reasonable given the potentially large expense involved with relocating facilities, changing suppliers, or reconfiguring networked systems. As an alternative, the reliability of existing infrastructure can be enhanced through efficient investments in protection and security measures. In light of recent world events, the identification of cost-effective protection strategies has been widely perceived as an urgent priority that demands not only greater public policy support (Sternberg and Lee [97]), but also the development of structured and analytical approaches (Jüttner et al. [52]). Planning for facility protection, in fact, is an enormous financial and logistical challenge if one considers the complexity of today's logistics systems, the interdependencies among critical infrastructures, the variety of threats and hazards, and the prohibitive costs involved in securing large

numbers of facilities. Despite the acknowledged need for analytical models able to capture these complexities, the study of mathematical models for allocation of protection resources is still in its infancy. The few *fortification models* that have been proposed in the literature are discussed in this section, together with possible extensions and variations.

4.2. Facility Location Models

Location models that explicitly address the issue of optimizing facility protection assume the existence of a supply system with P operating facilities. Facilities are susceptible to deliberate sabotage or accidental failures, unless protective measures are taken to prevent their disruption. Given limited protection resources, the models aim to identify the subset of facilities to protect to minimize efficiency losses due to intentional or accidental disruptions. Typical measures of efficiency are distance traveled, transportation cost, or captured demand.

4.2.1. Expected Cost Models. In this section, we present the P-median fortification problem (PMFP) (Scaparra [79]). This model builds on the well-known P-median problem (Hakimi [43, 44]). It assumes that the P facilities in the system have unlimited capacity and that the system users receive service from their nearest facility. As in the design model RFLP, each facility may fail or be disrupted with a fixed probability q. A disrupted facility becomes inoperable, so that the customers currently served by it must be reassigned to their closest nondisrupted facility. Limited fortification resources are available to protect Q of the P facilities. A protected facility becomes immune to disruption. The PMFP identifies the fortification strategy that minimizes the expected transportation costs.

The model definition builds on the notation used in the previous sections, with the exception that J now denotes the set of existing, rather than potential, facilities. Additionally, let i_k denote the kth closest facility to customer i, and let d_i^k be the expected transportation cost between customer i and its closest operational facility, given that the $k-1$ closest facilities to i are not protected, and the kth closest facility to i is protected. These expected costs can be calculated as follows.

$$d_i^k = \sum_{j=1}^{k-1} q^{j-1}(1-q)d_{ii_j} + q^{k-1}d_{ii_k} \tag{26}$$

The PMFP uses two sets of decision variables:

$$Z_j = \begin{cases} 1, & \text{if facility } j \text{ is fortified,} \\ 0, & \text{otherwise} \end{cases}$$

$$W_{ik} = \begin{cases} 1, & \text{if the } k-1 \text{ closest facilities to customer } i \text{ are not protected but the } k\text{th} \\ & \text{closest facility is,} \\ 0, & \text{otherwise.} \end{cases}$$

Then PMFP can be formulated as the following mixed integer program.

$$\text{(PMFP)} \quad \text{minimize} \quad \sum_{i\in I}\sum_{k=1}^{P-Q+1} h_i d_i^k W_{ik} \tag{27}$$

subject to

$$\sum_{k=1}^{P-Q+1} W_{ik} = 1 \quad \forall i \in I, \tag{28}$$

$$W_{ik} \le Z_{i_k} \quad \forall i \in I,\ k = 1, \ldots, P-Q+1 \tag{29}$$

$$W_{ik} \le 1 - Z_{i_{k-1}} \quad \forall\, i \in I,\ k = 2, \ldots, P - Q + 1 \tag{30}$$

$$\sum_{j \in J} Z_j = Q \tag{31}$$

$$W_{ik} \in \{0,1\} \quad \forall\, i \in I,\ k = 1, \ldots, P - Q + 1 \tag{32}$$

$$Z_j \in \{0,1\} \quad \forall\, j \in J \tag{33}$$

The objective function (27) minimizes the weighted sum of expected transportation costs. Note that the expected costs d_i^k and the variables W_{ik} need only be defined for values of k between 1 and $P - Q + 1$. In fact, in the worst case, the closest protected facility to customer i is its $(P - Q + 1)$st-closest facility. This occurs if the Q fortified facilities are the Q furthest facilities from i. If all of the $P - Q$ closest facilities to i fail, customer i is assigned to its $(P - Q + 1)$st-closest facility. Assignments to facilities that are further than the $(P - Q + 1)$st-closest facility will never be made in an optimal solution. For each customer i, constraints (28) force exactly one of the $P - Q + 1$ closest facilities to i to be its closest protected facility. The combined use of constraints (29) and (30) ensures that the variable W_{ik} that equals 1 is the one associated with the smallest value of k such that the kth closest facility to i is protected. Constraint (31) specifies that only Q facilities can be protected. Finally, constraints (32) and (33) represent the integrality requirements of the decision variables.

The PMFP is an integer programming model and can be solved with general purpose mixed-integer programming software. Possible extensions of the model include the cases in which facilities have different failure probabilities and fortification only reduces, but does not eliminate, the probability of failure. Unfortunately, (PMFP) cannot be easily adjusted to handle capacity restrictions. As for the design version of the problem, if the system facilities have limited capacities, explicit scenarios must be used to model possible disruption patterns. The capacitated version of (PMFP) can be formulated in an analogous way to the scenario-based model (CRFLP) discussed in §3.2.1.

(CPMFP) minimize $$\sum_{s \in S} q_s \sum_{i \in I} \sum_{j \in J} h_i d_{ij} Y_{ijs} \tag{34}$$

subject to

$$\sum_{j \in J} Y_{ijs} = 1 \quad \forall\, i \in I,\ s \in S \tag{35}$$

$$\sum_{i \in I} h_i Y_{ijs} \le (1 - a_{js}) b_j + a_{js} b_j Z_j \quad \forall\, j \in J,\ s \in S \tag{36}$$

$$\sum_{j \in J} Z_j = Q \tag{37}$$

$$X_j \in \{0,1\} \quad \forall\, j \in J \tag{38}$$

$$Y_{ijs} \in \{0,1\} \quad \forall\, i \in I,\ j \in J,\ s \in S \tag{39}$$

(CPMFP) uses the same parameters a_{js} and set S as (CRFLP) to model different scenarios. It also assumes that the set of existing facilities J is augmented with the unlimited-capacity emergency facility u. CPMFP differs from CRFLP only in a few aspects: No decisions must be made in terms of facility location, so the fixed cost for locating facilities are not included in the objective; the capacity constraints (36) must reflect that if a facility j is protected ($Z_j = 1$), then that facility remains operable (and can supply b_j units of demand) even in those scenarios s that assume its failure ($a_{js} = 1$). Finally, constraint (37) must be added to fix the number of possible fortifications.

Note that in both models (PMFP) and (CPMFP), the cardinality constraints (31) and (37) can be replaced by more general resource constraints to handle the problem in which

each facility requires a different amount of protection resources and there is a limit on the total resources available for fortification. Alternately, one could incorporate this cost into the objective function and omit the budget constraint. The difference between these two approaches is analogous to that between the P-median problem and the UFLP.

4.2.2. Worst-Case Cost Models. When modeling protection efforts, it is crucial to account for hazards to which a facility may be exposed. It is evident that protecting against intentional attacks is fundamentally different from protecting against acts of nature. Whereas nature hits at random and does not adjust its behavior to circumvent security measures, an intelligent adversary may adjust its offensive strategy depending on which facilities have been protected, for example, by hitting different targets. The expected cost models discussed in §4.2.1 do not take into account the behavior of adversaries and are, therefore, more suitable to model situations in which natural and accidental failures are a major concern. The models in this section have been developed to identify cost-effective protection strategies against malicious attackers.

A natural way of looking at fortification problems involving intelligent adversaries is within the framework of a leader-follower or Stackelberg game [96], in which the entity responsible for coordinating the fortification activity, or *defender*, is the leader and the attacker, or *interdictor*, is the follower. Stackelberg games can be expressed mathematically as bilevel programming problems (Dempe [31]): The upper-level problem involves decisions to determine which facilities to harden, whereas the lower-level problem entails the interdictor's response of which unprotected facilities to attack to inflict maximum harm. Even if in practice we cannot assume that the attacker is always able to identify the best attacking strategy, the assumption that the interdictor attacks in an optimal way is used as a tool to model worst-case scenarios and estimate worst-case losses in response to any given fortification strategy.

The worst-case cost version of PMFP was formulated as a bilevel program by Scaparra and Church [82]. The model, called the R-interdiction median model with fortification (RIMF), assumes that the system defender has resources to protect Q facilities, whereas the interdictor has resources to attack R facilities, with $Q + R < P$. In addition to the fortification variables Z_j defined in §4.2.1, the RIMF uses the following interdiction and assignment variables:

$$S_j = \begin{cases} 1, & \text{if facility } j \text{ is interdicted,} \\ 0, & \text{otherwise} \end{cases}$$

$$Y_{ij} = \begin{cases} 1, & \text{if customer } i \text{ is assigned to facility } j \text{ after interdiction,} \\ 0, & \text{otherwise.} \end{cases}$$

Additionally, the formulation uses the set $T_{ij} = \{k \in J \mid d_{ik} > d_{ij}\}$ defined for each customer i and facility j. T_{ij} represents the set of existing sites (not including j) that are farther than j is from demand i. The RIMF can then be stated mathematically as follows.

$$\text{(RIMF)} \qquad \text{minimize} \quad H(Z) \tag{40}$$

$$\text{subject to}$$

$$\sum_{j \in J} Z_j = Q \tag{41}$$

$$Z_j \in \{0,1\} \quad \forall j \in J, \tag{42}$$

where

$$H(Z) = \text{maximize} \sum_{i \in I} \sum_{j \in J} h_i d_{ij} Y_{ij} \tag{43}$$

$$\sum_{j \in J} Y_{ij} = 1 \quad \forall i \in I \tag{44}$$

$$\sum_{j \in J} S_j = R \tag{45}$$

$$\sum_{h \in T_{ij}} Y_{ih} \leq S_j \quad \forall i \in I,\ j \in J \tag{46}$$

$$S_j \leq 1 - Z_j \quad \forall j \in J \tag{47}$$

$$S_j \in \{0,1\} \quad \forall j \in J \tag{48}$$

$$Y_{ij} \in \{0,1\} \quad \forall i \in I,\ j \in J \tag{49}$$

In the above bilevel formulation, the leader allocates exactly Q fortification resources (41) to minimize the highest possible level of weighted distances or costs, H, (40) deriving from the loss of R of the P facilities. That H represents worst-case losses after the interdiction of R facilities is enforced by the follower problem, whose objective involves maximizing the weighted distances or service costs (43). In the lower-level interdiction problem (RIM; Church et al. [21]), constraints (44) state that each demand point must be assigned to a facility after interdiction. Constraint (45) specifies that only R facilities can be interdicted. Constraint (46) maintains that each customer must be assigned to its closest open facility after interdiction. More specifically, these constraints state that if a given facility j is not interdicted ($S_j = 0$), a customer i cannot be served by a facility further than j from i. Constraints (47) link the upper- and lower-level problems by preventing the interdiction of any protected facility. Finally, constraints (42), (48), and (49) represent the integrality requirements for the fortification, interdiction, and assignment variables, respectively. Note that the binary restrictions for the Y_{ij} variables can be relaxed, because an optimal solution with fractional Y_{ij} variables only occurs when there is a distance tie between two nondisrupted closest facilities to customer i. Such cases, although interesting, do not affect the optimality of the solution.

Church and Scaparra [18] and Scaparra and Church [81] demonstrate that it is possible to formulate (RIMF) as a single-level program and discuss two different single-level formulations. However, both formulations require the explicit enumeration of all possible interdiction scenarios and, consequently, their applicability is limited to problem instances of modest size. A more efficient way of solving (RIMF) is through the implicit enumeration scheme proposed by Scaparra and Church [82] and tailored to the bilevel structure of the problem.

A stochastic version of (RIMF), in which an attempted attack on a facility is successful only with a given probability, can be obtained by replacing the lower-level interdiction model (43)–(49) with the probabilistic R-interdiction median model introduced by Church and Scaparra [19].

Different variants of the RIMF model, aiming at capturing additional levels of complexity, are currently under investigation. Ongoing studies focus, for example, on the development of models and solution approaches for the capacitated version of the RIMF.

The RIMF assumes that at most R facilities can be attacked. Given the large degree of uncertainty characterizing the extent of man-made and terrorist attacks, this assumption should be relaxed to capture additional realism. An extension of (RIMF) that includes random numbers of possible losses as well as theoretical results to solve this expected loss version to optimality are currently under development.

Finally, bilevel fortification models similar to (RIMF) can be developed for protecting facilities in supply systems with different service protocols and efficiency measures. For example, in emergency service and supply systems, the effects of disruption may be better measured in terms of the reduction in operational response capability. In these problem settings, the most disruptive loss of R facilities would be the one causing the maximal drop in user

demand that can be supplied within a given time or distance threshold. This problem can be modeled by replacing the interdiction model (43)–(49) with the R-interdiction covering problem introduced by Church et al. [21] and by minimizing, instead of maximizing, the upper-level objective function H, which now represents the worst-case demand coverage decrease after interdiction.

4.3. Network Design Models

The literature dealing with the disruption of existing networked systems has primarily focused on the analysis of risk and vulnerabilities through the development of interdiction models. Interdiction models have been used by several authors to identify the most critical components of a system, i.e., those nodes or linkages that, if disabled, cause the greatest disruption to the flow of services and goods through the network. A variety of models, which differ in terms of objectives and underlying network structures, have been proposed in the interdiction literature. For example, the effect of interdiction on the maximum flow through a network is studied by Wollmer [105] and Wood [106]. Israeli and Wood [48] analyze the impact of link removals on the shortest path length between nodes. Lim and Smith [61] treat the multicommodity version of the shortest path problem, with the objective of assessing shipment revenue reductions due to arc interdictions. A review of interdiction models is provided by Church et al. [21].

Whereas interdiction models can help reveal potential weaknesses in a system, they do not explicitly address the issue of optimizing security. Scaparra and Cappanera [80] demonstrate that securing those network components that are identified as critical in an optimal interdiction solution will not necessarily provide the most cost-effective protection against disruptions. Optimal interdiction is a function of what is fortified, so it is important to capture this interdependency within a modeling framework. The models detailed in the next section explicitly addressed the issue of fortification in networked systems.

4.3.1. Expected Cost. In this section, we present the reliable network fortification problem (RNFP), which can be seen as the protection counterpart of the RNDP discussed in §3.3.1. The problem is formulated below by using the same notation as in §3.3.1 and the fortification variables $Z_j = 1$ if node j is fortified, and $Z_j = 0$ otherwise.

$$\text{(RNFP)} \quad \text{minimize} \quad \sum_{s\in S} q_s \sum_{(i,j)\in A} d_{ij} Y_{ijs} \tag{50}$$

subject to

$$\sum_{(j,i)\in A} Y_{jis} - \sum_{(i,j)\in A} Y_{ijs} = b_j \quad \forall\, j \in V\backslash\{u,v\},\ s \in S \tag{51}$$

$$\sum_{(j,i)\in A} Y_{jis} \le (1-a_{js})k_j + a_{js}k_j Z_j \quad \forall\, j \in V_0,\ s \in S \tag{52}$$

$$\sum_{j\in J} Z_j = Q \tag{53}$$

$$Z_j \in \{0,1\} \quad \forall\, j \in V_0 \tag{54}$$

$$Y_{ijs} \ge 0 \quad \forall\, (i,j) \in A,\ s \in S \tag{55}$$

The general structure of the RNFP and the meaning of most of its components are as in the RNDP. A difference worth noting is that now the capacity constraints (52) maintain that each fortified node preserves its original capacity in every failure scenario.

The RNFP can be easily modified to handle the problem in which fortification does not completely prevent node failures but only reduces the impact of disruptions. As an example, we can assume that a protected node only retains part of its capacity in case of failure and

that the level of capacity that can be secured depends on the amount of protective resources invested on that node. To model this variation, we denote by f_j the fortification cost incurred to preserve one unit of capacity at node j and by B the total protection budget available. Also, we define the continuous decision variables T_j as the level of capacity that is secured at node j (with $0 \le T_j \le k_j$). RNFP can be reformulated by replacing the capacity constraints (52) and the cardinality constraints (53) with the following two sets of constraints:

$$\sum_{(j,i)\in A} Y_{jis} \le (1 - a_{js})k_j + a_{js}T_j \quad \forall\, j \in V_0,\ s \in S \tag{56}$$

and

$$\sum_{j\in J} f_j T_j \le B. \tag{57}$$

4.3.2. Worst-Case Cost. The concept of protection against worst-case losses for network models has been briefly discussed by Brown et al. [14] and Salmeron et al. [77]. The difficulty in addressing this kind of problem is that their mathematical representation requires building tri-level optimization models, to represent fortification, interdiction, and network flow decisions. Multilevel optimization problems are not amenable to solution by standard mixed integer programming methodologies, and no universal algorithm exists for their solutions. To the best of our knowledge, the first attempt at modeling and solving network problems involving protection issues was undertaken by Scaparra and Cappanera [80], who discuss two different models: In the first model, optimal fortification strategies are identified to thwart as much as possible the action of an opponent who tries to disrupt the supply task from a supply node to a demand node by disabling or interdicting network linkages. This model is referred to as the shortest path interdiction problem with fortification (SPIF). In the second model, the aim is to fortify network components so as to maximize the flow of goods and services that can be routed through a supply network after a worst-case disruption of some of the network nodes or linkages. This model is referred to as the maximum flow interdiction problem with fortification (MFIF). The two multilevel models incorporate in the lower level the interdiction models described by Israeli and Wood [48] and by Wood [106], respectively.

In both models, there is a supply node o and a demand node d. Additionally, in the SPIF, each arc (i, j) has a penalty of p_{ij} associated with it that represents the cost increase to ship flow through it if the arc is interdicted. (The complete loss of an arc can be captured in the model by choosing p_{ij} sufficiently large.) In the MFIF, each arc has a penalty r_{ij} representing the percentage capacity reduction of the arc deriving from interdiction. (If $r_{ij} = 100\%$, then an interdicted arc (i, j) is completely destroyed.) The remaining notation used by the two models is the same as in §§3.3.1 and 4.3.1.

Note that in both models, it is assumed that the critical components that can be interdicted and protected are the network linkages. However, it is easy to prove that problems in which the critical components are the nodes can be reduced to critical arc models by opportunely augmenting the underlying graph (Corley and Chang [23]). Hence, we describe the more-general case of arc protection and interdiction.

The three-level SPIF can be formulated as follows.

$$\text{(SPIF)} \qquad \min_{Z\in F} \max_{S\in D} \min_{Y} \sum_{(i,j)\in A} (d_{ij} + p_{ij}S_{ij})Y_{ij} \tag{58}$$

subject to

$$\sum_{(j,i)\in A} Y_{ji} - \sum_{(i,j)\in A} Y_{ij} = b_j \quad \forall\, j \in V \tag{59}$$

$$S_{ij} \le 1 - Z_{ij} \quad \forall\, (i,j) \in A \tag{60}$$

$$Y_{ij} \ge 0 \quad \forall\, (i,j) \in A \tag{61}$$

where $F = \{Z \in \{0,1\}^n | \sum_{(i,j)\in A} Z_{ij} = Q\}$ and $D = \{S \in \{0,1\}^n | \sum_{(i,j)\in A} S_{ij} = R\}$. Also, as in standard shortest path problems, we define $b_o = 1, b_d = -1$, and $b_j = 0$ for all the other nodes j in V. The objective function (58) computes the minimum-cost path after the worst-case interdiction of R unprotected facilities. This cost includes the penalties associated with interdicted arcs. Protected arcs cannot be interdicted (60).

The MFIF model can be formulated in a similar way as follows.

$$\text{(MFIF)} \quad \max_{z\in F} \min_{s\in D} \max_{Y\geq 0} \quad W \tag{62}$$

subject to

$$\sum_{(j,i)\in A} Y_{ji} - \sum_{(i,j)\in A} Y_{ij} = W \quad j = o \tag{63}$$

$$\sum_{(j,i)\in A} Y_{ji} - \sum_{(i,j)\in A} Y_{ij} = 0 \quad \forall j \in V\backslash\{o,d\} \tag{64}$$

$$\sum_{(j,i)\in A} Y_{ji} - \sum_{(i,j)\in A} Y_{ij} = -W \quad j = d \tag{65}$$

$$Y_{ij} \leq k_{ij}(1 - r_{ij}S_{ij}) \quad \forall (i,j) \in A \tag{66}$$

(60)–(61)

In (MFIF), the objective (62) is to maximize the total flow W through the network after the worst-case interdiction of the capacities of R arcs. Capacity reductions due to interdiction are calculated in (66). Constraints (63)–(65) are standard flow conservation constraints for maximum-flow problems.

The two three-level programs (SPIF) and (MFIF) can be reduced to bilevel programs by taking the dual of the inner network flow problems. Scaparra and Cappanera [80] show how the resulting bilevel problem can be solved efficiently through an implicit enumeration scheme that incorporates network optimization techniques. The authors also show that optimal fortification strategies can be identified for relatively large networks (hundreds of nodes and arcs) in reasonable computational time and that significant efficiency gains (in terms of path costs or flow capacities) can be achieved even with modest fortification resources.

Model (MFIF) can be easily modified to handle multiple sources and multiple destinations. Also, a three-level model can be built along the same lines as (SPIF) and (MFIF) for multicommodity flow problems. For example, by embedding the interdiction model proposed in Lim and Smith [61] in the three-level framework, it is possible to identify optimal fortification strategies for maximizing the profit that can be obtained by shipping commodities across a network, while taking into account worst-case disruptions.

5. Conclusions

In this tutorial, we have attempted to illustrate the wide range of strategic planning models available for desiging supply chain networks under the threat of disruptions. A planner's choice of model will depend on a number of factors, including the type of network under consideration, the status of existing facilities in the network, the firm's risk preference, and the resources available for constructing, fortifying, and operating facilities.

We believe that several promising avenues exist for future research in this field. First, the models we discussed in this tutorial tend to be much more difficult to solve than their reliable-supply counterparts—most have significantly more decision variables, many have additional hard constraints, and some have multiple objectives. For these models to be implemented broadly in practice, better solution methods are required.

The models presented above consider the cost of reassigning customers or rerouting flow after a disruption. However, other potential repercussions should be modeled. For example,

firms may face costs associated with destroyed inventory, reconstruction of disrupted facilities, and customer attrition (if the disruption does not affect the firm's competitors). In addition, the competitive environment in which a firm operates may significantly affect the decisions the firm makes with respect to risk mitigation. For many firms, the key objective may be to ensure that their post-disruption situation is no worse than that of their competitors. Embedding these objectives in a game-theoretic environment is another important extension.

Finally, most of the existing models for reliable supply chain network design use some variation of a minimum-cost objective. Such objectives are most applicable for problems involving the distribution of physical goods, primarily in the private sector. However, reliability is critical in the public sector as well, for the location of emergency services, post-disaster supplies, and so on. In these cases, cost is less important than proximity, suggesting that coverage objectives may be warranted. The application of such objectives to reliable facility location and network design problems will enhance the richness, variety, and applicability of these models.

Acknowledgments

The authors gratefully acknowledge financial support from EPSRC (Ref. 320 21095), the Higher Education Funding Council for England (HEFCE), and the National Science Foundation (Grant DMI-0522725). The authors also thank Michael Johnson for his feedback on earlier drafts of this tutorial.

References

[1] Antonio Arreola-Risa and Gregory A. DeCroix. Inventory management under random supply disruptions and partial backorders. *Naval Research Logistics* 45:687–703, 1998.

[2] M. L. Balinski. Integer programming: Methods, uses, computation. *Management Science* 12(3):253–313, 1965.

[3] Alexei Barrionuevo and Claudia H. Deutsch. A distribution system brought to its knees. *New York Times* (Sept. 1) C1, 2005.

[4] R. Beach, A. P. Muhlemann, D. H. R. Price, A. Paterson, and J. A. Sharp. A review of manufacturing flexibility. *European Journal of Operational Research* 122:41–57, 2000.

[5] Emre Berk and Antonio Arreola-Risa. Note on "Future supply uncertainty in EOQ models." *Naval Research Logistics* 41:129–132, 1994.

[6] Oded Berman and Dimitri Krass. Facility location problems with stochastic demands and congestion. Zvi Drezner and H. W. Hamacher, eds. *Facility Location: Applications and Theory.* Springer-Verlag, New York, 331–373, 2002.

[7] O. Berman, M. J. Hodgson, and D. Krass. Flow-interception problems. Zvi Drezner, ed. *Facility Location: A Survey of Applications and Methods.* Springer Series in Operations Research, Springer, New York, 389–426, 1995.

[8] Oded Berman, Dmitry Krass, and Mozart B. C. Menezes. Facility reliability issues in network p-median problems: Strategic centralization and colocation effects. *Operations Research.* Forthcoming. 2005.

[9] Oded Berman, Dmitry Krass, and Mozart B. C. Menezes. MiniSum with imperfect information: Trading off quantity for reliability of locations. Working paper, Rotman School of Management, University of Toronto, Toronto, ON, Canada, 2005.

[10] Oded Berman, Richard C. Larson, and Samuel S. Chiu. Optimal server location on a network operating as an $M/G/1$ queue. *Operations Research* 33(4):746–771, 1985.

[11] D. E. Bienstock, E. F. Brickell, and C. L. Monma. On the structure of minimum-weight k-connected spanning networks. *SIAM Journal on Discrete Mathematics* 3:320–329, 1990.

[12] E. K. Bish, A. Muriel, and S. Biller. Managing flexible capacity in a make-to-order environment. *Management Science* 51(2):167–180, 2005.

[13] Ken Brack. Ripple effect from GM strike build. *Industrial Distribution* 87(8):19, 1998.

[14] G. G. Brown, W. M. Carlyle, J. Salmerón, and K. Wood. Analyzing the vulnerability of critical infrastructure to attack and planning defenses. H. J. Greenberg, ed., *Tutorials in Operations Research.* INFORMS, Hanover, MD, 102–123, 2005.

[15] Markus Bundschuh, Diego Klabjan, and Deborah L. Thurston. Modeling robust and reliable supply chains. Working paper, University of Illinois, Urbana-Champaign, IL, 2003.

[16] R. D. Carr, H. J. Greenberg, W. E. Hart, G. Konjevod, E. Lauer, H. Lin, T. Morrison, and C. A. Phillips. Robust optimization of contaminant sensor placement for community water systems. *Mathematical Programming* 107:337–356, 2005.

[17] Richard Church and Charles ReVelle. The maximal covering location problem. *Papers of the Regional Science Association* 32:101–118, 1974.

[18] Richard L. Church and Maria P. Scaparra. Protecting critical assets: The r-interdiction median problem with fortification. *Geographical Analysis*. Forthcoming. 2005.

[19] R. L. Church and M. P. Scaparra. Analysis of facility systems' reliability when subject to attack or a natural disaster. *Reliability and Vulnerability in Critical Infrastructure: A Quantitative Geographic Perspective.* A. T. Murray and T. H. Grubesic, eds. Springer-Verlag, New York, 2006.

[20] R. L. Church, M. P. Scaparra, and J. R. O'Hanley. Optimizing passive protection in facility systems. Working paper, ISOLDE X, Spain, 2005.

[21] Richard L. Church, Maria P. Scaparra, and Richard S. Middleton. Identifying critical infrastructure: The median and covering facility interdiction problems. *Annals of the Association of American Geographers* 94(3):491–502, 2004.

[22] C. Colbourn. *The Combinatorics of Network Reliability*. Oxford University Press, New York, 1987.

[23] H. W. Corley and H. Chang. Finding the most vital nodes in a flow network. *Management Science* 21(3):362–364, 1974.

[24] Mark S. Daskin. Application of an expected covering model to emergency medical service system design. *Decision Sciences* 13:416–439, 1982.

[25] Mark S. Daskin. A maximum expected covering location model: Formulation, properties and heuristic solution. *Transportation Science* 17(1):48–70, 1983.

[26] Mark S. Daskin. *Network and Discrete Location: Models, Algorithms, and Applications*. Wiley, New York, 1995.

[27] Mark S. Daskin, Collette R. Coullard, and Zuo-Jun Max Shen. An inventory-location model: Formulation, solution algorithm and computational results. *Annals of Operations Research* 110:83–106, 2002.

[28] M. S. Daskin, K. Hogan, and C. ReVelle. Integration of multiple, excess, backup, and expected covering models. *Environment and Planning B* 15(1):15–35, 1988.

[29] Mark S. Daskin, Lawrence V. Snyder, and Rosemary T. Berger. Facility location in supply chain design. A. Langevin and D. Riopel, eds., *Logistics Systems: Design and Operation.* Springer, New York, 39–66, 2005.

[30] A. de Toni and S. Tonchia. Manufacturing flexibility: A literature review. *International Journal of Production Research* 36(6):1587–1617, 1998.

[31] S. Dempe. *Foundations of Bilevel Programming*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002.

[32] Z. Drezner. Heuristic solution methods for two location problems with unreliable facilities. *Journal of the Operational Research Society* 38(6):509–514, 1987.

[33] Zvi Drezner, ed. *Facility Location: A Survey of Applications and Methods*. Springer-Verlag, New York, 1995.

[34] H. A. Eiselt, Michel Gendreau, and Gilbert Laporte. Location of facilities on a network subject to a single-edge failure. *Networks* 22:231–246, 1992.

[35] D. Elkins, R. B. Handfield, J. Blackhurst, and C. W. Craighead. 18 ways to guard against disruption. *Supply Chain Management Review* 9(1):46–53, 2005.

[36] B. Fortz and M. Labbe. Polyhedral results for two-connected networks with bounded rings. *Mathematical Programming Series A* 93:27–54, 2002.

[37] Justin Fox. A meditation on risk. *Fortune* 152(7):50–62, 2005.

[38] M. Garg and J. C. Smith. Models and algorithms for the design of survivable multicommodity flow networks with general failure scenarios. *Omega*. Forthcoming. 2006.

[39] M. Gendreau, G. Laporte, and I. Parent. Heuristics for the location of inspection stations on a network. *Naval Research Logistics* 47:287–303, 2000.

[40] Stephen C. Graves and Brian T. Tomlin. Process flexibility in supply chains. *Management Science* 49(7):907–919, 2003.

[41] M. Grötschel, C. L. Monma, and M. Stoer. Polyhedral and computational investigations for designing communication networks with high survivability requirements. *Operations Research* 43(6):1012–1024, 1995.

[42] Diwakar Gupta. The (Q, r) inventory system with an unreliable supplier. *INFOR* 34(2):59–76, 1996.

[43] S. L. Hakimi. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research* 12(3):450–459, 1964.

[44] S. L. Hakimi. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations Research* 13(3):462–475, 1965.

[45] Julia L. Higle. Stochastic programming: Optimization when uncertainty matters. *Tutorials in Operations Research.* INFORMS, Hanover, MD, 30–53, 2005.

[46] M. J. Hodgson, K. E. Rosing, and J. Zhang. Locating vehicle inspection stations to protect a transportation network. *Geographical Analysis* 28:299–314, 1996.

[47] Wallace J. Hopp and Zigeng Yin. Protecting supply chain networks against catastrophic failures. Working paper, Norhtwestern University, Evanston, IL, 2006.

[48] E. Israeli and R. K. Wood. Shortest-path network interdiction. *Networks* 40(2):97–111, 2002.

[49] J. C. James and S. Salhi. A Tabu Search heuristic for the location of multi-type protection devices on electrical supply tree networks. *Journal of Combinatorial Optimization* 6:81–98, 2002.

[50] Hyong-Mo Jeon, Lawrence V. Snyder, and Z. J. Max Shen. A location-inventory model with supply disruptions. Working paper, Lehigh University, Bethlehem, PA, 2006.

[51] William C. Jordan and Stephen C. Graves. Principles on the benefits of manufacturing process flexibility. *Management Science* 41(4):577–594, 1995.

[52] U. Jüttner, H. Peck, and M. Christopher. Supply chain risk management: Outlining an agenda for future research. *International Journal of Logistics: Research and Applications* 6(4): 197–210, 2003.

[53] Hyoungtae Kim, Jye-Chyi Lu, and Paul H. Kvam. Ordering quantity decisions considering uncertainty in supply-chain logistics operations. Working paper, Georgia Institute of Technology, Atlanta, GA, 2005.

[54] Anton J. Kleywegt, Alexander Shapiro, and Tito Homem-de-Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization* 12(2):479–502, 2001.

[55] Panagiotis Kouvelis and Gang Yu. *Robust Discrete Optimization and Its Applications.* Kluwer Academic Publishers, Boston, MA, 1997.

[56] Richard C. Larson. A hypercube queuing model for facility location and redistricting in urban emergency services. *Computers and Operations Research* 1:67–95, 1974.

[57] Richard C. Larson. Approximating the performance of urban emergency service systems. *Operations Research* 23(5):845–868, 1975.

[58] Almar Latour. Trial by fire: A blaze in Albuquerque sets off major crisis for cell-phone giants—Nokia handles supply chain shock with aplomb as Ericsson of Sweden gets burned—Was Sisu the difference? *Wall Street Journal* (Jan. 29) A1, 2001.

[59] S. D. Lee. On solving unreliable planar location problems. *Computers and Operations Research* 28:329–344, 2001.

[60] Devin Leonard. The only lifeline was the Wal-Mart. *Fortune* 152(7):74–80, 2005.

[61] C. Lim and J. C. Smith. Algorithms for discrete and continuous multicommodity flow network interdiction problems. *IIE Transactions.* Forthcoming. 2006.

[62] Jeff Linderoth, Alexander Shapiro, and Stephen Wright. The empirical behavior of sampling methods for stochastic programming. *Annals of Operations Research* 142:219–245, 2006.

[63] Barry C. Lynn. *End of the Line: The Rise and Coming Fall of the Global Corporation.* Doubleday, New York, 2005.

[64] Esmail Mohebbi. Supply interruptions in a lost-sales inventory system with random lead time. *Computers and Operations Research* 30:411–426, 2003.

[65] Esmail Mohebbi. A replenishment model for the supply-uncertainty problem. *International Journal of Production Economics* 87(1):25–37, 2004.

[66] C. L. Monma. Minimum-weight two-connected spanning networks. *Mathematical Programming* 46(2):153–171, 1990.

[67] C. L. Monma and D. F. Shalcross. Methods for designing communications networks with certain 2-connected survivability constraints. *Operations Research* 37:531–541, 4 1989.

[68] Jad Mouawad. Katrina's shock to the system. *New York Times* (Sept. 4) 3.1, 2005.

[69] J. R. O'Hanley and R. L. Church. Planning for facility-loss: A bilevel decomposition algorithm for the maximum covering location-interdiction problem. Working paper, Oxford University, Oxford, England, 2005.

[70] Susan Hesse Owen and Mark S. Daskin. Strategic facility location: A review. *European Journal of Operational Research* 111(3):423–447, 1998.

[71] Mahmut Parlar. Continuous-review inventory problem with random supply interruptions. *European Journal of Operational Research* 99:366–385, 1997.

[72] M. Parlar and D. Berkin. Future supply uncertainty in EOQ models. *Naval Research Logistics* 38:107–121, 1991.

[73] Hasan Pirkul. The uncapacitated facility location problem with primary and secondary facility requirements. *IIE Transactions* 21(4):337–348, 1989.

[74] Reuters. Lumber, coffee prices soar in Katrina's wake. *Reuters* (Sept. 1) 2005.

[75] Charles ReVelle and Kathleen Hogan. The maximum availability location problem. *Transportation Science* 23(3):192–200, 1989.

[76] J. B. Rice and F. Caniato. Building a secure and resilient supply network. *Supply Chain Management Review* 7(5):22–30, 2003.

[77] J. Salmeron, R. K. Wood, and R. Baldick. Analysis of electric grid security under terrorist threat. *IEEE Transactions on Power Systems* 19(2):905–912, 2004.

[78] Tjendera Santoso, Shabbir Ahmed, Marc Goetschalckx, and Alexander Shapiro. A stochastic programming approach for supply chain network design under uncertainty. *European Journal of Operational Research* 167:96–115, 2005.

[79] M. P. Scaparra. Optimal resource allocation for facility protection in median systems. Working paper, University of Kent, Canterbury, England, 2006.

[80] M. P. Scaparra and P. Cappanera. Optimizing security investments in transportation and telecommunication networks. INFORMS Annual Meeting, San Francisco, CA, 2005.

[81] Maria P. Scaparra and Richard L. Church. An optimal approach for the interdiction median problem with fortification. Working Paper 78, Kent Business School, Canterbury, England, UK, 2005.

[82] Maria P. Scaparra and Richard L. Church. A bilevel mixed integer program for critical infrastructure protection planning. *Computers and Operations Research.* Forthcoming. 2006.

[83] Alexander Shapiro and Tito Homem-de-Mello. A simulation-based approach to two-stage stochastic programming with recourse. *Mathematical Programming* 81:301–325, 1998.

[84] Yossi Sheffi. *The Resilient Enterprise: Overcoming Vulnerability for Competitive Advantage.* MIT Press, Cambridge, MA, 2005.

[85] Zuo-Jun Max Shen, Collette R. Coullard, and Mark S. Daskin. A joint location-inventory model. *Transportation Science* 37(1):40–55, 2003.

[86] D. R. Shier. *Network Reliability and Algebraic Structures.* Clarendon Press, Oxford, England, 1991.

[87] Martin L. Shooman. *Reliability of Computer Systems and Networks: Fault Tolerance, Analysis, and Design.* John Wiley & Sons, New York, 2002.

[88] Robert L. Simison. GM contains its quarterly loss at $809 million. *Wall Street Journal* (Oct. 14) A2, 1998.

[89] Robert L. Simison. GM says strike reduced its earnings by $2.83 billion in 2nd and 3rd periods. *Wall Street Journal* (Aug. 17) 1, 1998.

[90] Lawrence V. Snyder. Facility location under uncertainty: A review. *IIE Transactions* 38(7):537–554, 2006.

[91] Lawrence V. Snyder. A tight approximation for a continuous-review inventory model with supplier disruptions. Working paper, Lehigh University, Bethlehem, PA, 2006.

[92] Lawrence V. Snyder and Mark S. Daskin. Reliability models for facility location: The expected failure cost case. *Transportation Science* 39(3):400–416, 2005.

[93] Lawrence V. Snyder and Mark S. Daskin. Stochastic p-robust location problems. *IIE Transactions* 38(11):971–985, 2006.

[94] Lawrence V. Snyder and Mark S. Daskin. Models for reliable supply chain network design. Alan T. Murray and Tony H. Grubesic, eds. *Reliability and Vulnerability in Critical Infrastructure: A Quantitative Geographic Perspective*. Forthcoming. Springer, New York, 2006.

[95] Lawrence V. Snyder and Z. Max Shen. Disruptions in multi-echelon supply chains: A simulation study. Working paper, Lehigh University, 2005.

[96] H. Stackelberg. *The Theory of Market Economy*. Oxford University Press, Oxford, England, 1952.

[97] E. Sternberg and G. Lee. Meeting the challenge of facility protection for homeland security. *Journal of Homeland Security and Emergency Management* 3(1):1–19, 2006.

[98] Brian T. Tomlin. The impact of supply-learning on a firm's sourcing strategy and inventory investment when suppliers are unreliable. Working Paper OTIM-2005-05, Kenan-Flagler Business School, University of North Carolina, Chapel Hill, NC, 2005.

[99] Brian T. Tomlin. Selecting a disruption-management strategy for short life-cycle products: Diversification, contingent sourcing, and demand management. Working Paper OTIM-2005-09, Kenan-Flagler Business School, University of North Carolina, Chapel Hill, NC, 2005.

[100] Brian T. Tomlin. On the value of mitigation and contingency strategies for managing supply-chain disruption risks. *Management Science* 52(5):639–657, 2006.

[101] Brian T. Tomlin and Lawrence V. Snyder. Inventory management with advanced warning of disruptions. Working paper, Lehigh University, Bethlehem, PA, 2006.

[102] Brian Tomlin and Yimin Wang. On the value of mix flexibility and dual sourcing in unreliable newsvendor networks. Working paper, Kenan-Flagler Business School, University of North Carolina, Chapel Hill, NC, 2004.

[103] Nursen Ş. Ülker and Lawrence V. Snyder. A model for locating capacitated, unreliable facilities. Working paper, Lehigh University, Bethlehem, PA, 2005.

[104] Jerry R. Weaver and Richard L. Church. A median location model with nonclosest facility service. *Transportation Science* 19(1):58–74, 1985.

[105] R. Wollmer. Removing arcs from a network. *Operations Research* 12(6):934–940, 1964.

[106] R. K. Wood. Deterministic network interdiction. *Mathematical and Computer Modelling* 17(2):1–18, 1993.

TUTORIALS IN
OPERATIONS RESEARCH
INFORMS 2006

ISBN13 978-1-877640-20-9

Contributing Authors

Farid Alizadeh ("Semidefinite and Second-Order Cone Programming and Their Application to Shape-Constrained Regression and Density Estimation") is a member of faculty of management and Rutgers Center for Operations Research at Rutgers University. He received his Ph.D. from the Computer and Information Science Department of the University of Minnesota in 1991. He subsequently served as an NSF postdoctoral associate at the International Computer Science Institute at the University of California, Berkeley. His main area of research is mathematical programming, particularly semidefinite programming, for which he has helped establish its conceptual foundations.

Dimitris Bertsimas ("Robust and Data-Driven Optimization: Modern Decision Making Under Uncertainty") is the Boeing Professor of Operations Research at the Sloan School of Management and Codirector of the Operations Research Center at the Massachusetts Institute of Technology. He is a former area editor of *Operations Research* and associate editor of *Mathematics of Operations Research.* He has published widely, has coauthored three graduate-level textbooks, and has supervised over 35 Ph.D. students. He is a member of the National Academy of Engineering, and he has received several awards including the Erlang Prize, the SIAM Optimization Prize, the Presidential Young Investigator Award, and the Bodosaki Prize.

Gérard P. Cachon ("Game Theory in Supply Chain Analysis") is the Fred R. Sullivan Professor of Operations and Information Management at The Wharton School, University of Pennsylvania. His research interests are primarily in supply chain management. He is the Editor of *Manufacturing & Service Operations Management.*

Richard L. Church ("Planning for Disruptions in Supply Chain Networks") is a professor in the Geography Department at the University of California, Santa Barbara. He received his Ph.D. in environmental systems engineering at the Johns Hopkins University. His research interests include the delivery of public services, transportation and location modeling, geographical information systems science, and natural resource management. He is the author of roughly 175 articles and monographs. He currently serves on the editorial boards of *Geographical Analysis* and *Socio-Economic Planning Sciences.*

Mark S. Daskin ("Planning for Disruptions in Supply Chain Networks") is a professor at Northwestern University. He received his Ph.D. from the Massachusetts Institute of Technology in 1978. He is the author of roughly 50 journal papers as well as *Network and Discrete Location: Models, Algorithms and Applications.* He is a past editor-in-chief of *Transportation Science* and *IIE Transactions.* He currently serves as the President of INFORMS.

Jeffrey Keisler ("Enhance Your Own Research Productivity Using Spreadsheets") is an assistant professor of management science and information systems at the University of Massachusetts–Boston. He previously worked as a decision analyst at General Motors, Argonne National Laboratory, and Strategic Decisions Group. He received his Ph.D. in decision sciences from Harvard University and MBA from the University of Chicago. His research interests are in spreadsheet modeling, decision analysis, and R&D portfolio management.

Andrew E. B. Lim ("Model Uncertainty, Robust Optimization, and Learning") obtained his Ph.D. in systems engineering from the Australian National University in 1998. He has held research positions at the Chinese University of Hong Kong, the University of Maryland, College Park, and Columbia University. From 2001 to 2002, he was Assistant Professor in the IEOR Department at Columbia University and is currently Associate Professor in the IEOR Department at the University of California, Berkeley. He received an NSF CAREER Award in 2004. His research interests are in the areas of stochastic control and applications. He is currently an associate editor for the *IEEE Transactions on Automatic Control.*

Katta G. Murty ("Linear Equations, Inequalities, Linear Programs, and a New Efficient Algorithm") is a professor of industrial and operations engineering at the University of Michigan, Ann Arbor. He received an M.S. in statistics from the Indian Statistical Institute in 1957 and Ph.D. in operations research from the University of California, Berkeley, in 1968. His research interests are in operations research and its applications to complex real-world decision problems, and in studying human impacts on nature. His recent research contributions are in fast-descent algorithms for LP without using matrix inversion operations and in portfolio models based on statistical learning.

Serguei Netessine ("Game Theory in Supply Chain Analysis") is an assistant professor of operations and information management at The Wharton School, University of Pennsylvania. His research focuses on game-theoretic applications and decentralized decision making in product and service delivery systems. He received his Ph.D./M.S. degrees in operations management from the W.E. Simon School of Business, University of Rochester, and he also holds B.S./M.S. degrees in electrical engineering from Moscow Institute of Electronic Technology.

Warren B. Powell ("Approximate Dynamic Programming for Large-Scale Resource Allocation Problems") is a professor in the Department of Operations Research and Financial Engineering at Princeton University. He received his Ph.D. from Massachusetts Institute of Technology and is the founding director of the CASTLE Laboratory at Princeton University. At CASTLE, he has developed large-scale stochastic optimization models for freight transportation. He has published over 100 papers and collaborated with many transportation firms and military branches in the U.S. and Canada. An INFORMS fellow, his recent research focuses on scalable algorithms for industrial applications using machine learning and math programming.

Maria P. Scaparra ("Planning for Disruptions in Supply Chain Networks") is an assistant professor at Kent Business School, University of Kent, United Kingdom. She earned a master's degree in engineering-economic systems and operations research at Stanford University, and her Ph.D. in mathematics applied to economic decisions at the University of Pisa, Italy. Her research interests include combinatorial and network optimization, large-scale neighborhood search techniques, location analysis, and infrastructure and supply chain reliability.

J. George Shanthikumar ("Model Uncertainty, Robust Optimization, and Learning") is Professor of Industrial Engineering and Operations Research at the University of California, Berkeley. He received his Ph.D. in industrial engineering from the University of Toronto in 1979. His research interests include: integrated interdisciplinary decision making, model uncertainty and learning, production systems modeling and analysis, reliability, simulation, stochastic processes, and supply chain management. He has written and coauthored over 250 papers on these topics. He is coauthor of the books *Stochastic Models of Manufacturing Systems* and *Stochastic Orders and Their Applications.*

Z. J. Max Shen ("Model Uncertainty, Robust Optimization and Learning") is an assistant professor in the Department of Industrial Engineering and Operations Research at the University of California, Berkeley. He received his Ph.D. from Northwestern University in 2000. His research interests are in supply chain design and management, mechanism design, and decision making with limited information.

Lawrence V. Snyder ("Planning for Disruptions in Supply Chain Networks") is an assistant professor of industrial and systems engineering at Lehigh University and is codirector of Lehigh's Center for Value Chain Research. He received his Ph.D. from Northwestern University. His research interests include modeling and solving stochastic problems in supply chain management, facility location, and logistics, especially problems involving supply uncertainty. He has worked as a supply chain engineer and consultant for firms in a wide range of industries.

Aurélie Thiele ("Robust and Data-Driven Optimization: Modern Decision Making Under Uncertainty") is the P.C. Rossin Assistant Professor in the Department of Industrial and Systems Engineering at Lehigh University. Her research focuses on decision making under uncertainty with imperfect information, with applications in revenue management. In 2003, her work on robust optimization was awarded first prize in the George Nicholson Paper Competition organized by INFORMS. Her research on data-driven optimization is currently funded by the National Science Foundation. She holds an M.Sc. and Ph.D. in electrical engineering and computer science from Massachusetts Institute of Technology, and a "diplôme d'ingénieur" from the École Nationale Supérieure des Mines de Paris in France.

Huseyin Topaloglu ("Approximate Dynamic Programming for Large-Scale Resource Allocation Problems") is an assistant professor in the School of Operations Research and Industrial Engineering at Cornell University. He holds a B.Sc. in industrial engineering from Bogazici University of Istanbul and a Ph.D. in operations research from Princeton University. His research interests are stochastic programming, dynamic programming, and machine learning. He particularly focuses on the applications of approximate dynamic programming to large-scale problems arising from the freight transportation industry. His current work addresses revenue management as well.

Geert-Jan van Houtum ("Multiechelon Production/Inventory Systems: Optimal Policies, Heuristics, and Algorithms") is an associate professor in operations management at Technische Universiteit Eindhoven, The Netherlands. His research interests are in multiechelon production/inventory systems, system-focused inventory control of spare parts, life cycle costs of capital goods, and multiproduct capacitated production/inventory systems. His research builds on fundamentals of inventory and queueing theory, and is strongly motivated by real-life problems. He is involved in joint research projects with several international companies, and he is a board member of the *European Supply Chain Forum* and the *Service Logistics Forum.*

Janet M. Wagner ("Enhance Your Own Research Productivity Using Spreadsheets") is an associate professor of management science and information systems at the University of Massachusetts–Boston, where she recently completed five years as the Associate Dean of the College of Management. This year she is an ACE fellow, spending the year at the University of Albany. She received her Ph.D. in operations research from Massachusetts Institute of Technology. Her research interests are in spreadsheet modeling and applications of OR/MS in health care, tax policy, and manufacturing.