MATHEMATICAL PROGRAMMING

Structures and Algorithms

JEREMY F. SHAPIRO

8

MATHEMATICAL PROGRAMMING: Structures and Algorithms

JEREMY F. SHAPIRO

Alfred P. Sloan School of Management
Massachusetts Institute of Technology

A Wiley-Interscience Publication
JOHN WILEY & SONS
New York · Chichester · Brisbane · Toronto

Library of Congress Cataloging in Publication Data

Shapiro, Jeremy F 1939–
Mathematical programming.

"A Wiley-Interscience publication."
Includes bibliographical references.
1. Programming (Mathematics) I. Title.

QA402.5.S5 519.7 79-4478
ISBN 0-471-77886-9

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

To my parents

PREFACE

The field of mathematical programming has reached a mature, although by no means moribund, state. The 1950s saw the development of many of the basic results and techniques, such as the simplex method and the Kuhn-Tucker theorem, and the identification of many important research problems. The 1960s saw the specialization of the field into subfields including integer programming, nonlinear programming, dynamic programming, and so on, with researchers working exclusively within one subfield on the development of special-purpose algorithms.

The 1970s have seen the development of conceptual results facilitating the reintegration of the various subfields. Of particular importance in this regard is Lagrangean duality and the related topic of convex analysis, which has played a central role in our understanding of all types of mathematical programming problems. By contrast, we have probably reached a point of diminishing returns in the development of new algorithms, although integer and nonlinear programming problems remain difficult to solve.

Applications of mathematical programming models have proliferated in the 1970s, in large part because of the existence of effective computer codes for solving them. These applications often involve a mixture of mathematical programming problems; for example, a plant location model for a manufacturing firm can include a network optimization problem describing the distribution of its products, nonlinear functions describing demand for the products in various markets s functions of their selling prices, and a mixed integer programming model describing the timing, sizing, and locational options available for new plants. Thus practical applications as well as theoretical developments have provided an impetus toward the integration of mathematical programming.

The mature state of mathematical programming suggests that the time is appropriate to try to present in one volume most of the basic results about the structure of mathematical programming problems and descriptions of many of the algorithms for solving them. In addition, such a volume should contain a development of Lagrangean duality and convex analysis that relates these subjects to the spectrum of mathematical programming problems, thereby exposing their differences and similarities and indicating how complex model integration can be achieved. These are the goals of this book. The reader should be cautioned, however, that the mathematical development is uniformly terse and will sometimes require a careful reading of new material.

The wide coverage of topics should permit this book to be used as the basic text in a variety of intermediate and advanced courses in mathematical programming. The following are suggested one-quarter or one-semester courses based on various combinations of the chapters:

Linear Programming	Chapters 1 and 2
Network Optimization	Chapters 3 and 4
Combinatorial Optimization	Chapter 3, Chapter 5 (Sections 5.1 to 5.3), and Chapter 8
Large Scale Optimization	Chapters 5 and 6
Nonlinear Programming	Chapters 5 and 7
Convex Analysis and the Theory of Optimization	Chapters 5 and 6, Appendix A

The book can provide the basic results for a course on a particular subject. These could be supplemented by the instructor's own notes, specific papers from the literature, or another book, according to taste.

The content and organization of the book also serve as an aide to students and researchers in mathematical programming as the field continues to grow in the years ahead. There are currently two major directions of growth. One is the development of fields of pure and applied mathematics that use mathematical programming constructs as basic building blocks; for example, the field of combinatorics that has grown out of combinatorial optimization, or abstract theories of optimization and convex analysis involving more complex vector spaces than finite dimensional spaces. This book provides students and researchers in these new fields with a broad survey of the basic results from which the mathematical generalizations are taken.

A second area of growth results from the infiltration of mathematical programming into other scientific disciplines such as economics, statistics, and engineering design, to name only a few. Many new theoretical and applied research problems have been discovered as the result of the use of mathematical programming in these new fields; for example, studies resulting from the integration of forecasting models with mathematical programming models of the efficient production and distribution of the commodities being forecasted. Again, this book provides much of the background necessary to study the use of mathematical programming in new fields.

The ideas and research accomplishments of many people influenced the development and content of this book. George Dantzig's prodigious contribution to mathematical programming must be singled out as having profoundly influenced us all. Ralph Gomory's work had a particularly strong influence on me, especially his research on the use of group theory to analyze integer programming problems. My initiation into research is due in large part to Harvey Wagner, who served as my dissertation advisor and encouraged me to pursue an academic career. My personal viewpoint of mathematical programming was shaped most by direct interaction with and reading the papers of David Bell, Tony Fiacco, Marshall Fisher, Tony Gorry, Ellis Johnson, Art Geoffrion, Richard Grinold, Jerry Gould, Mike Held, Dick Karp, Tom Magnanti, George Nemhauser, Bill Northup, Herb Scarf, and Larry Wolsey.

Contracts from the Army Research Office and the National Science Foundation to the Operations Research Center at M.I.T. supported research that subsequently motivated me to write this book.

Tom Magnanti showed devotion beyond the call of duty by reading the final draft of this book and by making many insightful comments as well as catching numerous errors. I also profited by comments received from Gabriel Bitran, Dorothy Elliott, Bruce Faaland, Eduardo Modiano and Paulo Villela. The remaining errors are my own responsibility. The title of the book was suggested by an anonymous reviewer who I also wish to thank. Finally, a debt of gratitude is owed to Kathy Sumera for having done most of the typing of the final manuscript. Joan Kargel and Betsy Sherman did the typing of earlier drafts.

JEREMY F. SHAPIRO

Cambridge, Massachusetts
April 1979

MATHEMATICAL CONVENTIONS

Whenever possible, we will not define explicitly a vector as a row or column vector and use it in both senses in matrix multiplication. The dimensions of the matrix and the nature of the matrix multiplication, pre- or postmultiplication, will usually determine unambiguously the sense of the vector. Thus, we will indicate vector transposes only where it is necessary for clarity. We will use the notation G^c to denote the complement of a set G. The notation $[X]^c$, where X is a subset of R^n, denotes the convex hull of X. The empty set will be denoted by $\varnothing$. Occasionally, we will also use $\varnothing$ to denote a mathematical function. The context will make the meaning clear. Other conventions will be explained as they arise.

CONTENTS

1 LINEAR PROGRAMMING

1.1 INTRODUCTION

Linear programming is fundamental to the study of mathematical programming because many important applications are linear programming problems, linear programming approximations are used in the solution of nearly all mathematical programming problems, and concepts and insights derived from linear programming are the basis for much of the general theory of mathematical programming. In this chapter we consider the well-known simplex algorithm for solving linear programming problems. The algorithm has been a great practical success in solving real-life problems as well as providing a mathematical tool that is of paramount importance to all areas of mathematical programming.

1.2 THE SIMPLEX METHOD

The linear programming problem in the form we will study it in this chapter is

$$\begin{aligned} \min z &= \mathbf{cx} \\ \text{s.t. } \mathbf{Ax} &= \mathbf{b} \\ \mathbf{x} &\geq 0 \end{aligned} \tag{1.1}$$

where $\mathbf{A}$ is a $m \times n$ matrix of rank m, $\mathbf{c}$ is a $1 \times n$ vector, and $\mathbf{b}$ is an $m \times 1$ vector. The notation "s.t." is an abbreviation for "subject to." This linear programming problem is said to have m rows and n columns. The matrix $\mathbf{A}$ is called the coefficient matrix, $\mathbf{c}$ the cost vector, $\mathbf{b}$ the right-hand-side vector. The linear function $\mathbf{cx}$ is called the objective function. The $m \times 1$ vectors $\mathbf{a}_j$, which are the columns of $\mathbf{A}$ will sometimes be called activities. An $n \times 1$ vector $\mathbf{x}$ satisfying $\mathbf{Ax} = \mathbf{b}$, and $\mathbf{x} \geq \mathbf{0}$, is called a feasible solution, and a minimal cost feasible solution is called an optimal solution. The set $\{\mathbf{x} | \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ is called the *feasible region* and it is a convex set; properties of convex sets and functions are reviewed in Appendix A. If (1.1) does not have a feasible solution, we take the minimum to be $+\infty$. If there exists a sequence of feasible solutions to (1.1) with objective function cost approaching $-\infty$, we take the minimum to be $-\infty$.

The constraints of a linear programming problem can be presented originally in the inequality form and converted to the equality form by the addition of slack variables or surplus variables. Specifically, an inequality $\Sigma_{j=1}^{n} a_{ij}x_j \le b_i$ is converted to the equality $\Sigma_{j=1}^{n} a_{ij}x_j + s_i = b_i$ by the addition of the *slack* variable s_i that is constrained to be nonnegative. The reverse inequality $\Sigma_{j=1}^{n} a_{ij}x_{ij} \ge b_i$ is converted to $\Sigma_{j=1}^{n} a_{ij}x_j - s_i = b_i$ by the addition of the nonnegative *surplus* variable s_i. Similarly, a maximization problem can be converted to a minimization problem by the rule $\max cx = -\min - cx$; that is, we minimize $-cx$ and the optimal objective function value of our maximization problem is the negative of this quantity. Finally, if a variable x_j is unconstrained in sign, then we make the substitution $x_j = x_j^+ - x_j^-$ where x_j^+ and x_j^- are constrained to be nonnegative.

One of the major strengths of linear programming as a model and the simplex method for solving it is the fact that so many decision problems can be formulated and solved in exactly the form (1.1). This is in contrast to the nonlinear and integer programming problems studied in later chapters that can also be stated in general form but their effective optimization often requires the use of specific algorithms that exploit special structures of particular problems.

The simplex method for solving the linear programming problem (1.1) is derived from classical theory for characterizing the solutions $\mathbf{x}$ satisfying $\mathbf{Ax} = \mathbf{b}$ by transforming this system to a more convenient equivalent form. A system $\mathbf{A}'\mathbf{x} = \mathbf{b}'$ is said to be *equivalent* to $\mathbf{Ax} = \mathbf{b}$ if their solution sets are equal. The simplex method adapts the classical theory to take into account the nonnegativity constraints $\mathbf{x} \ge \mathbf{0}$ and to select a minimal cost solution from among the feasible solutions.

Let $\mathbf{B}$ denote an $m \times m$ nonsingular submatrix of $\mathbf{A}$; without loss of generality, assume we have reordered the columns of $\mathbf{A}$ so that $\mathbf{A} = (\mathbf{B}, \mathbf{N})$. The matrix $\mathbf{B}$ is called a *basis*. Let $\mathbf{x}$ be similarly partitioned as $(\mathbf{x_B}, \mathbf{x_N})$; the variables $\mathbf{x_B}$ are called *basic* variables and the $\mathbf{x_N}$ are called *nonbasic* variables. A characterization of the solutions to the system $\mathbf{Ax} = \mathbf{b}$ or equivalently $\mathbf{Bx_B} + \mathbf{Nx_N} = \mathbf{b}$ is given by the vectors $(\mathbf{x_B}, \mathbf{x_N})$ in R^n where $\mathbf{x_N}$ is any vector in R^{n-m} and

$$\mathbf{x_B} = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}\mathbf{x_N} \tag{1.2}$$

The solution $(\mathbf{x_B}, \mathbf{x_N}) = (\mathbf{B}^{-1}\mathbf{b}, 0)$ is called a *basic solution*. It should be clear that a basic solution is unique since $\mathbf{B}$ is a nonsingular matrix. A basis $\mathbf{B}$ is called a *feasible basis* if $\mathbf{B}^{-1}\mathbf{b} \ge 0$. We say a basic feasible solution is *degenerate* if one or more components $\bar{b}_i$ of the vector $\bar{\mathbf{b}} = \mathbf{B}^{-1}\mathbf{b}$ is zero. If all components $\bar{b}_i$ are positive, then the basic feasible solution is called *nondegenerate*.

It is necessary to study in detail the process whereby $\mathbf{Ax}=\mathbf{b}$ can be transformed with respect to a basis inverse to the equivalent form (1.2). Specifically, we want to study the elementary operations whereby the original system

$$\sum_{j=1}^{n} a_{ij}x_j = b_i \qquad i=1,\ldots,m \tag{1.3}$$

is transformed to the basic system

$$x_i + \sum_{j=m+1}^{n} \bar{a}_{ij}x_j = \bar{b}_i \qquad i=1,\ldots,m \tag{1.4}$$

The $\bar{a}_{ij}$ are the elements of the $m\times(n-m)$ matrix $\mathbf{B}^{-1}\mathbf{N}$ and the $\bar{b}_i$ are the elements of the m-vector $\mathbf{B}^{-1}\mathbf{b}$.

The transformation from (1.3) to (1.4) can be accomplished by a series of *pivot operations*, which we now define. We leave it to the reader to verify that a pivot operation on a linear system produces an equivalent system.

Definition 1.1. A pivot operation on a linear system consists of m elementary operations, which transform the system to an equivalent system in which a specified variable has a coefficient of unity in one equation and zero elsewhere. The specific operations are:

1. Select a term a_{rs} in the system (1.3) such that $a_{rs}\neq 0$. This term is called the *pivot* term.
2. Replace equation r by the equation r multiplied by $(1/a_{rs})$.
3. For $i=1,2,\ldots,m$, except $i=r$, replace equation i by the sum of equation i and the replaced equation r multiplied by $(-a_{is})$.

The transformation of (1.3) to the basic system (1.4) is accomplished by a sequence of m pivots. The first pivot term can be any a_{rs} such that $a_{rs}\neq 0$. After the first pivot operations has been completed, the second pivot term is selected using a nonzero element from any equation except r, say equation r^1. After the second pivot operation has been completed the third pivot term is selected in the resulting system from any equation except r and r^1. The general pivot operation is identical with the pivot term chosen from equations that do not correspond to equations previously selected. For simplicity, we have assumed that the form (1.4) can be achieved without interchanging rows or columns.

A fundamental property of the simplex method is the transformation from one basic system to another, which results when a basic variable is replaced by a nonbasic variable. This is accomplished by selecting any element $\bar{a}_{rs} \neq 0$ in (1.4) and pivoting on it. The result is

$$x_i + \frac{(-\bar{a}_{is})}{\bar{a}_{rs}} x_r + \sum_{\substack{j=m+1 \\ j \neq s}}^{n} \left(\bar{a}_{ij} - \frac{\bar{a}_{rj}}{\bar{a}_{rs}} \cdot \bar{a}_{is} \right) x_j = \bar{b}_i - \frac{\bar{b}_r}{\bar{a}_{rs}} \cdot \bar{a}_{is}$$

$$i = 1, \dots, m, \; i \neq r \tag{1.5}$$

$$\frac{1}{\bar{a}_{rs}} x_r + \sum_{\substack{j=m+1 \\ j \neq s}}^{n} \frac{\bar{a}_{rj}}{\bar{a}_{rs}} x_j + x_s = \frac{\bar{b}_r}{\bar{a}_{rs}}$$

The indicated basic solution obtained by setting the nonbasic variables to zero in (1.5) is

$$x_i = \bar{b}_i - \frac{\bar{b}_r}{\bar{a}_{rs}} \cdot \bar{a}_{is} \qquad i = 1, \dots, m, \; i \neq r$$

$$x_s = \frac{\bar{b}_r}{\bar{a}_{rs}} \tag{1.6}$$

To see that this pivoting operation is equivalent to substituting the column $\mathbf{a}_s$ for the column $\mathbf{a}_r$ in the basis, let

$$\mathbf{B}_0 = (\mathbf{a}_1, \dots, \mathbf{a}_{r-1}, \mathbf{a}_r, \mathbf{a}_{r+1}, \dots, \mathbf{a}_m)$$

and

$$\mathbf{B}_1 = (\mathbf{a}_1, \dots, \mathbf{a}_{r-1}, \mathbf{a}_s, \mathbf{a}_{r+1}, \dots, \mathbf{a}_m).$$

We can readily verify that

$$\mathbf{B}_1^{-1} = \mathbf{E}\mathbf{B}_0^{-1} \tag{1.7}$$

where

$$\mathbf{E} = \begin{bmatrix} 1 & & & & \frac{-\bar{a}_{1s}}{\bar{a}_{rs}} & & & \\ & 1 & & & & & & \\ & & \ddots & & \vdots & & & \\ & & & 1 & \frac{-\bar{a}_{r-1,s}}{\bar{a}_{rs}} & & & \\ & & & & \frac{1}{\bar{a}_{rs}} & & & \\ & & & & \frac{-\bar{a}_{r+1,s}}{\bar{a}_{rs}} & 1 & & \\ & & & & \vdots & & \ddots & \\ & & & & \frac{-\bar{a}_{ms}}{\bar{a}_{rs}} & & & 1 \end{bmatrix}$$

The non-unit column of $\mathbf{E}$ is column r. In particular, (1.7) follows by verifying $\mathbf{E}\mathbf{B}_0^{-1}\mathbf{a}_i=\mathbf{e}_i$ for $i=1,\ldots,m$, $i\neq r$, and $\mathbf{E}\mathbf{B}_0^{-1}\mathbf{a}_s=\mathbf{e}_r$ where $\mathbf{e}_i$ is the ith unit vector in R^m.

To complete the argument, note that (1.4) can be expressed as

$$\mathbf{B}_0^{-1}(\mathbf{A}\mathbf{x})=\mathbf{B}_0^{-1}\mathbf{b}$$

and (1.5) as

$$\mathbf{E}\mathbf{B}_0^{-1}(\mathbf{A}\mathbf{x})=\mathbf{E}\mathbf{B}_0^{-1}\mathbf{b}$$

where the latter equation can be verified by direct calculation. Thus, since $\mathbf{B}_1^{-1}=\mathbf{E}\mathbf{B}_0^{-1}$, pivoting on $\bar{a}_{rs}$ in (1.4) to obtain (1.5) is equivalent to changing the basis representation of $\mathbf{A}\mathbf{x}=\mathbf{b}$ with respect to $\mathbf{B}_0$ to one with respect to $\mathbf{B}_1$.

With this background, we show how the simplex method proceeds from a basic feasible solution to an optimal basic feasible solution. We will show later in this section how an initial basic feasible solution is obtained. The first result is a test for optimality of a basic feasible solution. Let the vector $\mathbf{c}$ be partitioned as $(\mathbf{c}_\mathbf{B},\mathbf{c}_\mathbf{N})$ conformally as $(\mathbf{x}_\mathbf{B},\mathbf{x}_\mathbf{N})$.

LEMMA 1.1. A basic feasible solution $(\mathbf{x}_\mathbf{B},\mathbf{x}_\mathbf{N})=(\mathbf{B}^{-1}\mathbf{b},\mathbf{0})$ to the linear programming problem (1.1) is a minimal cost solution if

$$\bar{\mathbf{c}}_\mathbf{N}=\mathbf{c}_\mathbf{N}-\mathbf{c}_\mathbf{B}\mathbf{B}^{-1}\mathbf{N}\geq\mathbf{0} \tag{1.8}$$

PROOF. We use (1.2) to substitute for the dependent basic variables $\mathbf{x}_\mathbf{B}$ in the objective function

$$z=\mathbf{c}_\mathbf{B}\mathbf{x}_\mathbf{B}+\mathbf{c}_\mathbf{N}\mathbf{x}_\mathbf{N}=\mathbf{c}_\mathbf{B}\mathbf{B}^{-1}\mathbf{b}+(\mathbf{c}_\mathbf{N}-\mathbf{c}_\mathbf{B}\mathbf{B}^{-1}\mathbf{N})\mathbf{x}_\mathbf{N}=\mathbf{c}_\mathbf{B}\mathbf{B}^{-1}\mathbf{b}+\bar{\mathbf{c}}_\mathbf{N}\mathbf{x}_\mathbf{N}$$

The cost of the basic feasible solution with $\mathbf{x}_\mathbf{N}=\mathbf{0}$ is $\mathbf{c}_\mathbf{B}\mathbf{B}^{-1}\mathbf{b}$. If $\mathbf{c}_\mathbf{N}-\mathbf{c}_\mathbf{B}\mathbf{B}^{-1}\mathbf{N}\geq\mathbf{0}$, then clearly $\mathbf{c}\mathbf{x}\geq\mathbf{c}_\mathbf{B}\mathbf{B}^{-1}\mathbf{b}$ for any feasible solution $(\mathbf{x}_\mathbf{B},\mathbf{x}_\mathbf{N})$. ■

Lemma 1.1 gives a sufficient condition for optimality of a basic feasible solution. The coefficients

$$\bar{c}_j=c_j-\mathbf{c}_\mathbf{B}\mathbf{B}^{-1}\mathbf{a}_j, \qquad j=1,\ldots,n$$

are called *reduced cost coefficients*; note that the reduced cost coefficient of a basic variable is 0 since $\mathbf{B}^{-1}\mathbf{a}_j$ is simply a unit vector that selects the coefficient c_j from the vector $\mathbf{c}_\mathbf{B}$. If we define the m-vector $\mathbf{u}=\mathbf{c}_\mathbf{B}\mathbf{B}^{-1}$, then each reduced cost coefficient $\bar{c}_j$ is derived from the original cost coefficient c_j by subtracting the quantity $\sum_{i=1}^{m}u_ia_{ij}$. In the next chapter, we give an economic interpretation of these u_i, which are called *shadow prices*. For future reference, we note from the proof of Lemma 1.1 that the objective function cost as a function of the reduced cost coefficients is

$$z=\bar{z}_0+\sum_{j=1}^{n}\bar{c}_jx_j \tag{1.9}$$

where $\bar{z}_0=\mathbf{c}_\mathbf{B}\mathbf{B}^{-1}\mathbf{b}$ (recall the $\bar{c}_j$ are zero for the basic x_j).