Springer Series in Operations Research

Editors:
Peter W. Glynn   Stephen M. Robinson

Springer
New York
Berlin
Heidelberg
Hong Kong
London
Milan
Paris
Tokyo

George B. Dantzig Mukund N. Thapa

Linear Programming

2: Theory and Extensions

With 45 Illustrations

Springer

Professor George B. Dantzig
Department of Management Science
and Engineering
Computer Science Department
Stanford University
Stanford, CA 94305
USA
George@LinearProgram.com

Dr. Mukund N. Thapa
President and CEO
Optical Fusion Inc.
P.O. Box 60478
Palo Alto, CA 94306-0478
President and CEO
Stanford Business Software, Inc.
P.O. Box 60398
Palo Alto, CA 94306-0398
Consulting Professor
Department of Management Science
and Engineering
Stanford University
Stanford, CA 94305
USA
Mukund@LinearProgram.com

Series Editors:

Peter W. Glynn
Department of Management Science
and Engineering
Terman Engineering Center
Stanford University
Stanford, CA 94305-4026
USA
glynn@leland.stanford.edu

Stephen M. Robinson
Department of Industrial Engineering
University of Wisconsin–Madison
1513 University Avenue
Madison, WI 53706-1572
USA
smrobins@facstaff.wisc.edu

The Library of Congress has catalogued the first volume as follows
Library of Congress Cataloging-in-Publication Data
Dantzig, George Bernard, 1914–
Linear programming 1 : introduction / George B. Dantzig
& Mukund N. Thapa.
p. cm. — (Springer series in operations research)
Includes bibliographical references and index.
ISBN 0-387-94833-3 (hardcover : alk. paper)
1. Linear programming. I. Thapa, Mukund Narain-Dhami. II. Title.
III. Series.
T57.74.D365 1997
619.7′2—dc20 96-36411

ISBN 0-387-98613-8 Printed on acid-free paper.

Printed in the United States of America.

9 8 7 6 5 4 3 2 1 SPIN 10689385

Typesetting: Pages created by the authors using LaTeX/TeX.

www.springer-ny.com

Springer-Verlag New York Berlin Heidelberg
A member of BertelsmannSpringer Science+Business Media GmbH

ABOUT THE AUTHORS

George B. Dantzig received the National Medal of Science from the President of the United States "for inventing Linear Programming and for discovering the Simplex Algorithm that led to wide-scale scientific and technical applications to important problems in logistics, scheduling, and network optimization, and to the use of computers in making efficient use of the mathematical theory." He is world famous for his twin discoveries; linear programming and the Simplex Algorithm, which together have enabled mankind for the first time to structure and solve extremely complex optimal allocation and resource problems. Among his other discoveries are the Decomposition Principle (with Philip Wolfe) which makes it possible to decompose and solve extremely large linear programs having special structures, and applications of these techniques with sampling to solving practical problems subject to uncertainty.

Since its discovery in 1947, the field of linear programming, together with its extensions (mathematical programming), has grown by leaps and bounds and is today the most widely used tool in industry for planning and scheduling.

George Dantzig received his master's from Michigan and his doctorate in mathematics from Berkeley in 1946. He worked for the U.S. Bureau of Labor Statistics, served as chief of the Combat Analysts Branch for USAF Headquarters during World War II, research mathematician for RAND Corporation, and professor and head of the Operations Research Center at the University of California, Berkeley. He is currently professor of Management Science and Engineering and Computer Science at Stanford University. He served as director of the System Optimization Laboratory and the PILOT Energy-Economic Model Project. Professor Dantzig's seminal work has laid the foundation for the field of systems engineering, which is widely used in network design and component design in computer, mechanical, and electrical engineering. His work inspired the formation of the Mathematical Programming Society, a major section of the Society of Industrial and Applied Mathematics, and numerous professional and academic bodies. Generations of Professor Dantzig's students have become leaders in industry and academia.

He is a member of the prestigious National Academy of Science, the American Academy of Arts and Sciences, and the National Academy of Engineering.

Mukund N. Thapa is the President & CEO of Optical Fusion, Inc., President of Stanford Business Software, Inc., as well as a consulting professor of Management Science and Engineering at Stanford University. He received a bachelor of technology degree in metallurgical engineering from the Indian Institute of Technology, Bombay, and M.S. and Ph.D. degrees in operations research from Stanford University in 1981. His Ph.D. thesis was concerned with developing specialized algorithms for solving large-scale unconstrained nonlinear minimization problems. By profession he is a software developer who produces commercial software products as well as commercial-quality custom software. Since 1978, Dr. Thapa has been applying the theory of operations research, statistics, and computer science to develop efficient, practical, and usable solutions to a variety of problems.

At Optical Fusion, Inc., Dr. Thapa is developing a multi-point IP-based videoconferencing system for use over networks. The feature-rich system will focus primarily on the needs of users and allow corproate users to seamlessly integrate conferencing in everyday business interactions. At Stanford Business Software, Dr. Thapa, ensures that the company produces high-quality turnkey software for clients. His expert knowledge of user friendly interfaces, data bases, computer science, and modular software design plays an important role in making the software practical and robust. His speciality is the application of numerical analysis methodology to solve mathematical optimization problems. He is also an experienced modeler who is often asked by clients to consult, prepare analyses, and to write position papers. At the Department of Management Science and Engineering, from time to time, Dr. Thapa teaches graduate-level courses in mathematical programming computation and numerical methods of linear programming.

TO

Tobias and Anja Dantzig, my parents, *in memoriam*
Anne S. Dantzig, my wife, and to
the great pioneers that made this field possible:
Wassily Leontief, Tjalling Koopmans, John von Neumann,
Albert Tucker, William Orchard-Hays, Martin Beale.

— George B. Dantzig

Radhika H. Thapa, my wife,
Isha, my daughter, and to
Devi Thapa & Narain S. Thapa, my parents.

— Mukund N. Thapa

Contents

List of Figures

List of Tables

PREFACE

Linear Programming 2 continues where *Linear Programming 1* left off. We assume that the reader has an introductory knowledge of linear programming, for example has read *Linear Programming 1: Introduction* (or its equivalent) and has knowledge of linear algebra (reviewed in the appendices in *Linear Programming 1*). In this volume, we prove all theorems stated and those that were sketched but not proved in *Linear Programming 1*, and we describe various extensions.

Linear Programming 2 is intended to be an advanced graduate text as well as a reference. Portions of *Linear Programming 1* and *Linear Programming 2* have been used in a graduate-level course that we have taught together. The rest of the discussion here summarizes the contents of this volume.

OUTLINE OF CHAPTERS

Chapter 1 (Geometry): In this chapter we study the geometry and properties of linear inequality systems and how they are related to the Simplex Method, which can be described as a movement along the edges of a convex polyhedral set to obtain a global minimum of the objective function, generate a class of feasible solutions for which the objective $z \to -\infty$, or determine that the convex polyhedral set is infeasible. The important separating hyperplane concepts are also discussed and proved.

Chapter 2 (Duality and Theorems of the Alternatives): We provide proofs for the *Weak* and *Strong* Duality Theorems. This is followed by additional theorems on duality; that is, the *Unboundedness Theorem* and the *Primal/Dual Optimality Criteria*. The chapter also discusses complementary slackness and various Theorems of the Alternatives: Gordan's Theorem, Farkas's Lemma, Stiemke's Theorem, Motzkin's Transposition Theorem, Ville's Theorem, and Tucker's Strict Complementary Slackness Theorem.

Chapter 3 (Early Interior-Point Methods): In this chapter we trace the early development of interior-point methods. The earliest known method is that attributable to von Neumann [1948], followed by Frisch [1957] (only referenced here), and Dikin [1967]. A theoretical breakthrough was due to Khachian [1979] who devloped a polynomial-time ellipsoid algorithm (only referenced here). This was followed by Karmarkar's [1984] polynomial-time interior-point algorithm.

Chapter 4 (Interior-Point Methods): Since the development of Karmarkar's [1984] algorithm several new important practical interior-point algorithms emerged. Among these are the primal logarithmic barrier method, primal-affine algorithm, dual logarithmic barrier method, dual-affine algorithm, and the primal-dual algorithm. All these algorithms are described. The optimal solution obtained by an interior-point method is not necessarily at a vertex; we describe a technique to make it into a vertex.

Chapter 5 (Degeneracy): When degeneracy occurs, it is possible for the Simplex Algorithm to have an infinite sequence of iterations with no decrease in the value of z. The chapter illustrates this with examples due to Hoffman, Beale, and Kuhn. Then various methods for resolving degeneracy are presented: Dantzig's Inductive Methods, Wolfe's Rule, Bland's Rule, and Krishna's Extra Column Rule. This is followed by a technique that attempts to avoid degenerate pivot by making use of an extra objective function and resultant reduced cost calculation.

Chapter 6 (Variants of the Simplex Method): Over the years several variants of the Simplex Algorithm have been proposed as a way to reduce the number of iterations. We start by describing an efficient way of determining an incoming column that yields the maximum improvement per iteration. Next we describe the Dual-Simplex Method, Parametric Linear Programming, Self-Dual Parametric Algorithm, Primal-Dual Algorithm, and a Phase I Least-Squares Algorithm.

Chapter 7 (Transportation Problem and Variations): The Classical Transportation Problem is stated, and various theorems are proved about it. An example is provided for cycling under degeneracy when the most negative reduced cost is used to select an incoming column. This is followed by a discussion of the Transshipment Problem and transportation problems with bounded partial sums.

Chapter 8 (Network Flow Theory): Theorems are proved about the Maximal-Flow problem and the Shortest-Route problem.

Chapter 9 (Generalized Upper Bounds): In this chapter we discuss a variation of the Simplex Algorithm to efficiently solve linear programs that have upper bounds on subsets of variables such that each variable appears in at most one subset. Such constraints are called *generalized upper bounds*.

Chapter 10 (Decomposition): Decomposition is a term to describe breaking a problem into smaller parts and then using a variant of the Simplex Algorithm to solve the enire problem efficiently. The chapter starts by describing Wolfe's Generalized Linear Program (or a linear program with variable coefficients). The Dantzig-Wolfe Decomposition Principle is described for solving this class of problems. This is followed by a description of Benders Decomposition which is the Dantzig-Wolfe Decomposition applied to the dual. Benders Decomposition is used to solve Stochastic Programs. Next we describe the application of Dantzig-Wolfe Decomposition to solving of Block-Angular systems. Then staircase structured problems are described; we show how to solve such problems using Dantzig-Wolfe Decomposition and Benders Decomposition. Finally, the possible use of decomposition to solve central planning problems is described.

Chapter 11 (Stochastic Programming Introduction): Here we introduce the concept of planning under uncertainty. Simple problems with uncertain demand and uncertain costs respectively are illustrated. This is followed by a discussion of the convexity property of multi-stage problems.

Chapter 12 (Two-Stage Stohastic Programs): An important class of optimization problems arise in dynamic systems that describe activities initiated at time t that have coefficients at time t and time $t+1$. Such problems, called *dynamic linear programs*, typically have a nonzero submatrix with a staircase structure. The simplest dynamic linear program has only two stages; this is discussed in this chapter.

Appendix A (Probability Theory Overview): In this appendix we introduce some basic concepts and notation of probability theory for use in solving stochastic linear programs.

LINEAR PROGRAMMING 1.

In a graduate course that we have taught together at Stanford, portions of *Linear Programming 1: Introduction* and *Linear Programming 2: Theory & Extensions* have been used.

Professor George B. Dantzig	Dr. Mukund N. Thapa
Stanford, California	Palo Alto, California
USA	USA

DEFINITION OF SYMBOLS

The notation described below will be followed in general. There may be some deviations where appropriate.

- Uppercase letters will be used to represent matrices.
- Lowercase letters will be used to represent vectors.
- All vectors will be column vectors unless otherwise noted.
- Greek letters will typically be used to represent scalars.

$\Re^n$	–	Real space of dimension n.
c	–	Coefficients of the objective function.
A	–	Coefficient matrix of the linear program.
B	–	Basis matrix (nonsingular). It contains the basic columns of A.
N	–	Nonbasic columns of A.
x	–	Solution of the linear program (typically the current one).
x_B	–	Basic solution (typically the current one).
x_N	–	Nonbasic solution (typically the current one).
(x, y)	–	The column vector consisting of components of the vector x followed by the components of y. This helps in avoiding notation such as $(x^T, y^T)^T$.
L	–	Lower triangular matrix with 1s on the diagonal.
U	–	Upper triangular matrix (sometimes R will be used).
R	–	Alternative notation for an upper triangular matrix.
D	–	Diagonal matrix.
$\text{Diag}\,(d)$	–	Diagonal matrix. Sometimes $\text{Diag}\,(d_1, d_2, \ldots, d_n)$ will be used.
D_x	–	$\text{Diag}\,(x)$.
I	–	Identity matrix.

e_j	–	jth column of an identity matrix.
e	–	Vector of 1s (dimension will be clear from the context).
E_j	–	Elementary matrix (jth column is different from the identity).
$\|\|v\|\|$	–	The 2-norm of a vector v; i.e., $\|\|v\|\|_2 = \sqrt{v^T v}$.
$\|\|v\|\|_1$	–	The 1-norm of a vector v; i.e., $\|\|v\|\|_1 = \sum_{i=1}^{n} \|v_i\|$.
$\|\|v\|\|_\infty$	–	The ∞-norm of a vector v; i.e., $\|\|v\|\|_\infty = \max_{i=1,\ldots,n} \|v_i\|$.
$\|\|A\|\|$	–	The 2-norm of an $m \times n$ matrix A; i.e., $\|\|A\|\|_2 = \sqrt{\lambda_{\max}(A^T A)}$.
$\|\|A\|\|_1$	–	The 1-norm of an $m \times n$ matrix A; i.e., $\|\|A\|\|_1 = \max_{j=1,\ldots,n} \sum_{i=1}^{m} \|a_{ij}\|$.
$\|\|A\|\|_\infty$	–	The ∞-norm of an $m \times n$ matrix A; i.e., $\|\|A\|\|_\infty = \max_{i=1,\ldots,m} \sum_{j=1}^{n} \|a_{ij}\|$.
$\det(A)$	–	Determinant of the matrix A.
$A_{\bullet j}$	–	jth column of A.
$A_{i \bullet}$	–	ith row of A.
B^t	–	The matrix B at the start of iteration t.
$B[t]$	–	Alternative form for the matrix B^t.
$\bar{B}$	–	Update from iteration t to iteration $t+1$.
B^{-1}_{ij}	–	Element (i, j) of B^{-1}.
$X \subset Y$	–	X is a proper subset of Y.
$X \subseteq Y$	–	X is a subset of Y.
$X \cup Y$	–	Set union, that is, the set $\{\omega \mid \omega \in X \text{ or } \omega \in Y\}$.
$X \cap Y$	–	The set $\{\omega \mid \omega \in X \text{ and } \omega \in Y\}$.
$X \setminus Y$	–	Set difference, that is, the set $\{\omega \mid \omega \in X, \omega \notin Y\}$
$\emptyset$	–	Empty set.
$\mid$	–	Such that. For example, $\{\, x \mid Ax \le b \,\}$ means the set of all x such that $Ax \le b$ holds.
α^n	–	A scalar raised to power n.
$(A)^n$	–	A square matrix raised to power n.
A^T	–	Transpose of the matrix A.
$\approx$	–	Approximately equal to.
$\gg$ ($\ll$)	–	Much greater (less) than.
$\succ$ ($\prec$)	–	Lexicographically greater (less) than.
$\leftarrow$	–	Store in the computer the value of the quantity on the right into the location where the quantity on the left is stored. For example, $x \leftarrow x + \alpha p$.
$O(v)$	–	Implies a number $\le kv$, where k, a fixed constant independent of the value of v, is meant to convey the the notion that k is some small integer value less than 10 (or possibly less than 100) and not something ridiculous like $k = 10^{100}$.

$\operatorname{argmin}_x f(x)$	–	The value of x where $f(x)$ takes on its global minimum value.
$\operatorname{argmin}_i \beta_i$	–	The value of the least index i where β_i takes on its minimum value.
LP	–	Linear program.
$\operatorname{sign}(\alpha)$	–	The sign of α. It is $+1$ if $\alpha \geq 0$ and -1 if $\alpha < 0$.