

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.

© 2003 George B. Dantzig and Mukund N. Thapa

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer-Verlag New York, Inc., 175 Fifth Avenue, New York, NY 10010, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

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 video-conferencing system for use over networks. The feature-rich system will focus primarily on the needs of users and allow corporate 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

List of Figures

1-1	Vector $(x_1, x_2)^T$	2
1-2	Examples of Two-Dimensional Convex Regions	3
1-3	Additional Examples of Two-Dimensional Convex Regions	3
1-4	Example of an Unbounded Two-Dimensional Convex Set and an Unbounded Three-Dimensional Convex Set	4
1-5	Examples of Two-Dimensional Nonconvex Regions	4
1-6	Points Common to Convex Sets	5
1-7	Example of a Line Segment	6
1-8	A Three-Dimensional Simplex	13
1-9	Local and Global Minima	14
1-10	Extreme Points	15
1-11	Joining Extreme Points	18
1-12	Geometrically the Iterates of the Simplex Algorithm Move Along the Edges of the Convex Set	21
1-13	Geometric Picture of the Distance of a Point to a Boundary	22
1-14	Movement of θ	24
1-15	Geometrically a Linear Program is a Center-of-Gravity Problem	26
1-16	Simplex Associated with an Iteration of the Simplex Algorithm ($m = 2$)	27
1-17	Geometry of the Simplex Algorithm for the Product Mix Problem	28
1-18	Simplex Associated with an Iteration of the Simplex Algorithm ($m = 3$)	30
1-19	Hyperplane H_j and Simplex \mathcal{S}	40
2-1	Illustration of the Duality Gap	45
2-2	Find Basic Feasible Solutions of Dual of Two Variable Primal	63
3-1	The Two-Dimensional Center-of-Gravity Problem: Find a Simplex that Contains the Origin	73
3-2	Finding an Improved Approximation	74
3-3	Convergence under Existence of a Ball \mathcal{B}	77
3-4	Degenerate Two-Dimensional Case	79
3-5	Decreasing x_s^t to Improve the Rate of Convergence	81
3-6	Ellipsoid Centered at $(1, 2)$	86
3-7	Ellipsoid Subproblem Centered at y^t	87

3-8	Comparison of a Move from a Point \hat{x}^t Near the “Center” Versus a Point \bar{x}^t Near the Boundary	101
3-9	Bound for $\hat{c}^T y^{t+1}$	105
4-1	Barrier Function Method: Approach of $x^*(\mu)$ to x^*	129
7-1	Network Representation of the Transportation Problem	209
7-2	Example of Standard Transportation Array	209
7-3	Cycling in the Transportation Problem	216
7-4	Perturbing the Transportation Problem	217
7-5	Example of a Standard Transshipment Array	219
7-6	The Transshipment Problem	221
8-1	A Simple Directed Network	232
8-2	Decomposition of Flows	234
8-3	Original Network and Associated Network with θ Flow	236
8-4	Example to Show Matching	245
8-5	Data for a Max-Flow Problem	246
10-1	Illustration of the Resolution Theorem	283
12-1	Benders Decomposition Applied to EQ-LP	339

List of Tables

2-1	Tucker Diagram (Partitioned)	54
5-1	Hoffman's Example of Cycling (Continued on the Right)	150
5-2	Hoffman's Example of Cycling (Continued from the Left)	151
5-3	Beale's Example of Cycling	152
5-4	Inductive Method and Wolfe's Rule Applied to Beale's Example . . .	157
6-1	Primal-Dual Correspondences	175
6-2	Primal-Simplex and Dual-Simplex Methods	180
9-1	An Example of GUB Constraints and Key Basic Variables	254
9-2	An Example of a Reordered Basis for GUB Constraints	254
12-1	Data for the Grape Grower's Dilemma	364
12-2	Data for Dinner Set Production Schedule	364

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.

\mathbb{R}^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 $\text{Diag}(d_1, d_2, \dots, d_n)$ will be used.
D_x	– $\text{Diag}(x)$.
I	– Identity matrix.

e_j	– j th column of an identity matrix.
e	– Vector of 1s (dimension will be clear from the context).
E_j	– Elementary matrix (j th 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,\dots,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,\dots,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,\dots,m} \sum_{j=1}^n a_{ij} $.
$\det(A)$	– Determinant of the matrix A .
$A_{\bullet j}$	– j th column of A .
$A_{i\bullet}$	– i th 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_{ij}^{-1}	– 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.
$ $	– Such that. For example, $\{x \mid Ax \leq b\}$ means the set of all x such that $Ax \leq b$ 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 $\leq kv$, where k , a fixed constant independent of the value of v , is meant to convey 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 $\alpha < 0$.



<http://www.springer.com/978-0-387-98613-5>

Linear Programming 2
Theory and Extensions
Dantzig, G.B.; Thapa, M.N.
2003, XXVI, 448 p., Hardcover
ISBN: 978-0-387-98613-5