

# 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  $\text{\LaTeX}/\text{\TeX}$ .

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 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

# Contents

<b>LIST OF FIGURES</b>	<b>xv</b>
<b>LIST OF TABLES</b>	<b>xvii</b>
<b>PREFACE</b>	<b>xix</b>
<b>DEFINITION OF SYMBOLS</b>	<b>xxiii</b>
<b>1 GEOMETRY OF LINEAR INEQUALITY SYSTEMS AND THE SIMPLEX METHOD</b>	<b>1</b>
1.1 CONVEXITY AND LINEAR INEQUALITY SYSTEMS . . . . .	1
1.1.1 Affine & Convex Combinations . . . . .	1
1.1.2 Two-dimensional Convex Regions . . . . .	3
1.1.3 Line Segments, Rays, and Half Lines . . . . .	5
1.1.4 General Convex Regions . . . . .	6
1.1.5 Hyperplanes and Half-Spaces . . . . .	7
1.1.6 Convexity of Half Spaces and Hyperplanes . . . . .	8
1.1.7 Convexity of the Set of Feasible Solutions of an LP . . . . .	9
1.1.8 Convex Polyhedrons, Polytopes, and Cones . . . . .	9
1.1.9 Separating Hyperplane . . . . .	11
1.2 SIMPLEX DEFINED . . . . .	13
1.3 GLOBAL MINIMUM, EXTREME POINTS, AND EDGES . . . . .	14
1.4 THE SIMPLEX METHOD VIEWED AS THE STEEPEST DE- SCENT ALONG EDGES . . . . .	20
1.5 THE SIMPLEX INTERPRETATION OF THE SIMPLEX METHOD	24
1.6 NOTES & SELECTED BIBLIOGRAPHY . . . . .	31
1.7 PROBLEMS . . . . .	31
<b>2 DUALITY AND THEOREMS OF THE ALTERNATIVES</b>	<b>43</b>
2.1 THE DUALITY THEOREM . . . . .	43
2.2 ADDITIONAL THEOREMS ON DUALITY . . . . .	47
2.2.1 Unboundedness Theorem . . . . .	47
2.2.2 Miscellaneous Theorems for the Standard Form . . . . .	48

2.3	COMPLEMENTARY SLACKNESS . . . . .	49
2.4	THEOREMS OF THE ALTERNATIVES . . . . .	50
2.4.1	Gordan's Theorem . . . . .	51
2.4.2	Farkas's Lemma . . . . .	52
2.4.3	Stiemke's Theorem . . . . .	53
2.4.4	Motzkin's Transposition Theorem . . . . .	54
2.4.5	Ville's Theorem . . . . .	55
2.4.6	Tucker's Strict Complementary Slackness Theorem . . . . .	56
2.5	NOTES & SELECTED BIBLIOGRAPHY . . . . .	58
2.6	PROBLEMS . . . . .	59
<b>3</b>	<b>EARLY INTERIOR-POINT METHODS</b>	<b>67</b>
3.1	VON NEUMANN'S METHOD . . . . .	70
3.1.1	The von Neumann Algorithm . . . . .	73
3.1.2	Improving the Rate of Convergence . . . . .	81
3.1.3	Von Neumann Algorithm as a Variant of the Simplex Algorithm	83
3.2	DIKIN'S METHOD . . . . .	84
3.2.1	Dikin's Algorithm . . . . .	87
3.2.2	Convergence of Dikin's Algorithm . . . . .	89
3.3	KARMAKAR'S METHOD . . . . .	100
3.3.1	Development of the Algorithm . . . . .	100
3.3.2	Proof of Convergence . . . . .	105
3.3.3	The Algorithm Summarized . . . . .	114
3.3.4	Converting a Standard LP to a Starting Form for the Algorithm	115
3.3.5	Computational Comments . . . . .	116
3.3.6	Complexity of von Neumann versus Karmarkar Algorithms . . . . .	118
3.4	NOTES & SELECTED BIBLIOGRAPHY . . . . .	119
3.5	PROBLEMS . . . . .	121
<b>4</b>	<b>INTERIOR-POINT METHODS</b>	<b>123</b>
4.1	NEWTON'S METHOD . . . . .	123
4.2	THE LINEAR LEAST-SQUARES PROBLEM . . . . .	127
4.3	BARRIER FUNCTION METHODS . . . . .	128
4.3.1	The Logarithmic Barrier Function . . . . .	128
4.3.2	Properties of Barrier Function Methods . . . . .	130
4.4	THE PRIMAL LOGARITHMIC BARRIER METHOD FOR SOLVING LINEAR PROGRAMS . . . . .	131
4.4.1	Details of the Method . . . . .	131
4.4.2	Initial Feasible Solution . . . . .	134
4.5	PRIMAL-DUAL LOGARITHMIC BARRIER METHODS . . . . .	134
4.6	RECOVERING A BASIC FEASIBLE SOLUTION . . . . .	137
4.7	COMPUTATIONAL COMMENTS . . . . .	139
4.8	NOTES & SELECTED BIBLIOGRAPHY . . . . .	140
4.9	PROBLEMS . . . . .	146



<b>5</b>	<b>DEGENERACY</b>	<b>149</b>
5.1	EXAMPLES OF CYCLING . . . . .	149
5.2	ON RESOLVING DEGENERACY . . . . .	153
5.3	DANTZIG'S INDUCTIVE METHOD . . . . .	154
5.4	WOLFE'S RULE . . . . .	156
5.5	BLAND'S RULE . . . . .	158
5.6	KRISHNA'S EXTRA COLUMN RULE . . . . .	160
5.7	ON AVOIDING DEGENERATE PIVOTS . . . . .	164
5.8	NOTES & SELECTED BIBLIOGRAPHY . . . . .	166
5.9	PROBLEMS . . . . .	167
<b>6</b>	<b>VARIANTS OF THE SIMPLEX METHOD</b>	<b>173</b>
6.1	INTRODUCTION . . . . .	173
6.2	MAX IMPROVEMENT PER ITERATION . . . . .	176
6.3	DUAL-SIMPLEX METHOD . . . . .	179
6.4	PARAMETRIC LINEAR PROGRAMS . . . . .	183
6.4.1	Parameterizing the Objective Function . . . . .	183
6.4.2	Parameterizing the Right-Hand Side . . . . .	187
6.5	SELF-DUAL PARAMETRIC ALGORITHM . . . . .	188
6.6	THE PRIMAL-DUAL ALGORITHM . . . . .	191
6.7	THE PHASE I LEAST-SQUARES ALGORITHM . . . . .	197
6.8	NOTES & SELECTED BIBLIOGRAPHY . . . . .	200
6.9	PROBLEMS . . . . .	202
<b>7</b>	<b>TRANSPORTATION PROBLEM AND VARIATIONS</b>	<b>207</b>
7.1	THE CLASSICAL TRANSPORTATION PROBLEM . . . . .	207
7.1.1	Mathematical Statement . . . . .	208
7.1.2	Properties of the System . . . . .	208
7.2	FINDING AN INITIAL SOLUTION . . . . .	213
7.3	FINDING AN IMPROVED BASIC SOLUTION . . . . .	214
7.4	DEGENERACY IN THE TRANSPORTATION PROBLEM . . . . .	216
7.5	TRANSSHIPMENT PROBLEM . . . . .	219
7.5.1	Formulation . . . . .	219
7.5.2	Reduction to the Classical Case by Computing Minimum Cost Routes . . . . .	222
7.5.3	Reduction to the Classical Case by the Transshipment Pro- cedure . . . . .	222
7.6	TRANSPORTATION PROBLEMS WITH BOUNDED PARTIAL SUMS . . . . .	225
7.7	NOTES & SELECTED BIBLIOGRAPHY . . . . .	227
7.8	PROBLEMS . . . . .	228

<b>8</b>	<b>NETWORK FLOW THEORY</b>	<b>231</b>
8.1	THE MAXIMAL FLOW PROBLEM . . . . .	232
8.1.1	Decomposition of Flows . . . . .	233
8.1.2	The Augmenting-Path Algorithm for Maximal Flow . . . . .	234
8.1.3	Cuts in a Network . . . . .	239
8.2	SHORTEST ROUTE . . . . .	241
8.3	MINIMUM COST-FLOW PROBLEM . . . . .	242
8.4	NOTES & SELECTED BIBLIOGRAPHY . . . . .	243
8.5	PROBLEMS . . . . .	245
<b>9</b>	<b>GENERALIZED UPPER BOUNDS</b>	<b>251</b>
9.1	PROBLEM STATEMENT . . . . .	251
9.2	BASIC THEORY . . . . .	253
9.3	SOLVING SYSTEMS WITH GUB EQUATIONS . . . . .	253
9.4	UPDATING THE BASIS AND WORKING BASIS . . . . .	257
9.5	NOTES & SELECTED BIBLIOGRAPHY . . . . .	264
9.6	PROBLEMS . . . . .	264
<b>10</b>	<b>DECOMPOSITION OF LARGE-SCALE SYSTEMS</b>	<b>265</b>
10.1	WOLFE'S GENERALIZED LINEAR PROGRAM . . . . .	267
10.2	DANTZIG-WOLFE (D-W) DECOMPOSITION PRINCIPLE . . . . .	280
10.2.1	D-W Principle . . . . .	284
10.2.2	D-W Decomposition Algorithm and Variants . . . . .	289
10.2.2.1	The D-W Algorithm . . . . .	289
10.2.2.2	Variants of the D-W Algorithm . . . . .	290
10.2.3	Optimality and Dual Prices . . . . .	290
10.2.4	D-W Initial Solution . . . . .	291
10.2.5	D-W Algorithm Illustrated . . . . .	292
10.3	BENDERS DECOMPOSITION . . . . .	299
10.3.1	Dual of D-W Decomposition . . . . .	299
10.3.2	Derivation of Benders Decomposition . . . . .	300
10.4	BLOCK-ANGULAR SYSTEM . . . . .	306
10.5	STAIRCASE STRUCTURED PROBLEMS . . . . .	308
10.5.1	Using Benders Decomposition . . . . .	309
10.5.2	Using D-W Decomposition . . . . .	310
10.5.3	Using D-W Decomposition with Alternate Stages Forming the Subproblems . . . . .	312
10.6	DECOMPOSITION USED IN CENTRAL PLANNING . . . . .	313
10.7	NOTES & SELECTED BIBLIOGRAPHY . . . . .	315
10.8	PROBLEMS . . . . .	317

<b>11 STOCHASTIC PROGRAMMING: INTRODUCTION</b>	<b>323</b>
11.1 OVERVIEW . . . . .	324
11.2 UNCERTAIN COSTS . . . . .	326
11.2.1 Minimum Expected Costs . . . . .	326
11.2.2 Minimum Variance . . . . .	327
11.3 UNCERTAIN DEMANDS . . . . .	329
11.4 NOTES & SELECTED BIBLIOGRAPHY . . . . .	332
11.5 PROBLEMS . . . . .	332
<b>12 TWO-STAGE STOCHASTIC PROGRAMS</b>	<b>335</b>
12.1 THE DETERMINISTIC TWO-STAGE LP PROBLEM . . . . .	335
12.2 THE ANALOGOUS STOCHASTIC TWO-STAGE LP PROBLEM . . . . .	336
12.3 LP EQUIVALENT OF THE STOCHASTIC PROBLEM (EQ-LP) . . . . .	337
12.3.1 LP Equivalent Formulation . . . . .	337
12.3.2 Geometric Description of Benders Decomposition Algorithm . . . . .	338
12.3.3 Decomposition Algorithm . . . . .	341
12.3.4 Theory behind the Algorithm . . . . .	348
12.4 SOLVING STOCHASTIC TWO-STAGE PROBLEMS USING SAM- PLING . . . . .	349
12.4.1 Overview . . . . .	349
12.4.2 Naive Sampling . . . . .	350
12.4.3 Sampling Methodology . . . . .	351
12.4.4 Estimating Upper Bound $z_{UB}$ for Min $z$ given $x = x^k$ . . . . .	351
12.4.5 Estimating Lower Bound $z_{LB}$ for Min $z$ . . . . .	352
12.5 USE OF IMPORTANCE SAMPLING . . . . .	354
12.5.1 Crude (Naive) Monte Carlo Methods . . . . .	355
12.5.2 Monte Carlo Methods using Importance Sampling . . . . .	356
12.6 NOTES & SELECTED BIBLIOGRAPHY . . . . .	360
12.7 PROBLEMS . . . . .	362
<b>A PROBABILITY THEORY: OVERVIEW</b>	<b>367</b>
A.1 BASIC CONCEPTS, EXPECTED VALUE, AND VARIANCE . . . . .	367
A.2 NORMAL DISTRIBUTION AND THE CENTRAL LIMIT THEO- REM . . . . .	370
A.3 CHI-SQUARE DISTRIBUTION, STUDENT'S $t$ -DISTRIBUTION, AND CONFIDENCE INTERVALS . . . . .	373
A.3.1 Chi-Square Distribution . . . . .	373
A.3.2 Student's $t$ -Distribution . . . . .	375
A.3.3 Confidence Intervals . . . . .	376
A.4 NOTES & SELECTED BIBLIOGRAPHY . . . . .	377
<b>REFERENCES</b>	<b>379</b>
<b>INDEX</b>	<b>439</b>

# 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 $\theta$ . . . . .	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 $\theta$ 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

# 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  $z \rightarrow -\infty$ , 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 developed 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 entire 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 Stochastic 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  
Stanford, California  
USA

Dr. Mukund N. Thapa  
Palo Alto, California  
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.

$\mathfrak{R}^n$	– 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 $\infty$ -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 $\infty$ -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.
$\alpha^n$	– 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  $\beta_i$  takes on its minimum value.
- LP – Linear program.
- $\operatorname{sign}(\alpha)$  – The sign of  $\alpha$ . It is  $+1$  if  $\alpha \geq 0$  and  $-1$  if  $\alpha < 0$ .