

**APPLIED MATHEMATICAL PROGRAMMING  
USING ALGEBRAIC SYSTEMS**

by

**Bruce A. McCarl**  
**Professor of Agricultural Economics**  
**Texas A&M University**  
[mccarl@tamu.edu](mailto:mccarl@tamu.edu)

**Thomas H. Spreen**  
**Professor of Food and Resource Economics**  
**University of Florida**

Preface .....	0-1
<b>CHAPTER I: INTRODUCTION .....</b>	<b>1-1</b>
1.1 Practical Problem Analysis .....	1-1
1.2 Mathematical Programming Approach .....	1-1
1.3 Mathematical Programming in Use .....	1-2
1.3.1 Generating Problem Insight .....	1-3
1.3.2 Numerical Mathematical Programming .....	1-3
1.3.3 Algorithmic Development .....	1-5
1.4 Book Plan .....	1-5
<b>CHAPTER II: LINEAR PROGRAMMING .....</b>	<b>2-1</b>
2.1 The Basic LP Problem .....	2-1
2.2 Basic LP Example .....	2-2
2.3 Other Forms of the LP Problem .....	2-4
2.4 Assumptions of LP .....	2-4
2.4.1 Objective Function Appropriateness .....	2-5
2.4.2 Decision Variable Appropriateness .....	2-5
2.4.3 Constraint Appropriateness .....	2-5
2.4.4 Proportionality .....	2-6
2.4.5 Additivity .....	2-7
2.4.6 Divisibility .....	2-7
2.4.7 Certainty .....	2-8
<b>CHAPTER III SOLUTION OF LP PROBLEMS: A MATRIX ALGEBRA APPROACH ..</b>	<b>3-1</b>
3.1 Matrix Formulation of the Linear Programming Problem .....	3-1
3.2 Solving LP's by Matrix Algebra .....	3-2
3.2.1 The Simplex Algorithm .....	3-7
3.2.2 Example .....	3-7
3.3 Solutions and Their Interpretation .....	3-11
3.3.1 General Solution Interpretation .....	3-11
3.3.2 Examples of Solution Interpretation .....	3-12
3.3.3 Finding Limits of Interpretation .....	3-14
3.3.3.1 Ranging Right-Hand-Sides .....	3-14
3.3.3.2 Ranging Objective Function Coefficients .....	3-17
3.3.3.3 Changes in the Technical Coefficient Matrix .....	3-18
3.3.4 Finding the Solution .....	3-20
3.3.5 Alternative Optimal and Degenerate Cases .....	3-21
3.3.6 Finding Shadow Prices for Bounds on Variables .....	3-23
3.4 Further Details on the Simplex Method .....	3-24
3.4.1 Updating the Basis Inverse .....	3-25
3.4.2 Advanced Bases .....	3-26
3.4.3 Finding an Initial Feasible Basis .....	3-27
3.4.3.1 BIG M Method .....	3-29

3.4.3.2 Phase I/Phase II Method .....	3-30
3.4.4 The Real LP Solution Method .....	3-31
References .....	3-33
CHAPTER IV: DUALITY IN LINEAR PROGRAMMING .....	4-1
4.1 Basic Duality .....	4-1
4.2 Primal-Dual Solution Inter-Relationships .....	4-3
4.2.1 Objective Function Interrelationships .....	4-4
4.2.2 Constructing Dual Solutions .....	4-5
4.2.3 Complementary Slackness .....	4-6
4.2.4 Zero Profits .....	4-7
4.2.5 Finding the Dual Solution Information .....	4-8
4.3 Duality Under Other Model Forms .....	4-8
4.4 The Character of Dual Solutions .....	4-12
4.5 Degeneracy and Shadow Prices .....	4-13
4.6 Primal Columns are Dual Constraints .....	4-14
References .....	4-15
CHAPTER V: LINEAR PROGRAMMING MODELING .....	5-1
5.1 Introduction to Algebraic Modeling .....	5-2
5.1.1 Defining Indices and GAMS Sets .....	5-2
5.1.2 Writing Equations .....	5-3
5.1.2.1 Unindexed Equations .....	5-3
5.1.2.2 Indexed Equations .....	5-4
5.1.3 Data Definitions .....	5-5
5.1.4 Entering Data Through Calculation .....	5-7
5.1.5 Specification of Variables .....	5-7
5.1.6 Equations .....	5-8
5.1.7 Writing Model Equations .....	5-8
5.1.7.1 Objective Function Equations .....	5-9
5.1.7.2 Constraint Equations .....	5-9
5.1.8 Specifying Bounds .....	5-10
5.1.9 Forming and Solving a Model .....	5-11
5.1.10 Report Writing .....	5-11
5.1.11 Additional GAMS Topics .....	5-12
5.2 Putting GAMS and Algebraic Modeling Together -- The Resource Allocation Problem .....	5-13
5.2.1 Example and GAMS Implementation .....	5-14
5.2.2 Model Solution .....	5-18
5.2.3 Comments .....	5-18
5.3 The Transportation Problem .....	5-19
5.3.1 Example .....	5-22
5.3.2 Comments .....	5-25
5.4 Diet/Feed Mix/Blending Problem .....	5-27
5.4.1 Example .....	5-30
5.4.2 Comments .....	5-31

5.5 Joint Products .....	5-32
5.5.1 Example .....	5-35
5.5.2 Comments .....	5-37
5.6 Comments on GAMS and Algebraic Modeling .....	5-38
5.6.1 Why Use Algebraic Modeling? .....	5-38
5.6.2 Why Use GAMS to Support Modeling? .....	5-39
References .....	5-44
CHAPTER VI: TOWARD PROPER MODELING .....	6-1
6.1 Structural Component Identification .....	6-1
6.1.1 Development of Model Constraints .....	6-1
6.1.2 Avoiding Improper Constraint Specifications .....	6-3
6.1.3 Variable Identification .....	6-4
6.1.4 Objective Function .....	6-5
6.1.5 Development of Model Structure .....	6-5
6.2 The Proper Usage of Variables and Constraints .....	6-6
6.2.1 Improper and Proper Handling of Joint Products .....	6-7
6.2.2 Alternatives for the Use of a Product .....	6-8
6.2.3 Improper Specification of Multiple Factor Relationships .....	6-9
6.2.4 Erroneous Imperfect Substitution .....	6-10
6.3 Simple Structural Checking .....	6-11
6.3.1 Homogeneity of Units .....	6-11
6.3.2 Numerical Model Analysis .....	6-12
6.4 Data Development .....	6-12
6.4.1 Time Frame .....	6-12
6.4.2 Uncertainty .....	6-13
6.4.3 Data Sources .....	6-13
6.4.4 Calculation Methods .....	6-13
6.4.5 Consistency .....	6-14
6.4.6 Specification of Individual Components .....	6-14
6.4.6.1 Objective Function Coefficients .....	6-14
6.4.6.2 Right hand Side Coefficients .....	6-15
6.4.6.3 Technical Coefficients .....	6-16
6.5 Purposeful Modeling .....	6-16
6.5.1 Model Structure .....	6-16
6.5.2 Report Writing .....	6-18
References .....	6-20
CHAPTER VII: MORE LINEAR PROGRAMMING MODELING .....	7-1
7.1 Assembly Problem .....	7-1
7.1.1 Example .....	7-2
7.1.2 Comments .....	7-3
7.2 Disassembly Problems .....	7-3
7.2.1 Example .....	7-4
7.2.2 Comments .....	7-4

7.3 Assembly-Disassembly .....	7-5
7.3.1 Example .....	7-6
7.3.2 Comments .....	7-7
7.4 Sequencing Problems .....	7-9
7.4.1 Example 1 .....	7-13
7.4.2 Example 2 .....	7-14
7.4.3 Comments .....	7-15
7.5 Storage Problems .....	7-16
7.5.1 Example .....	7-18
7.5.2 Comments .....	7-18
7.6 Input Output Analysis .....	7-19
7.6.1 Example .....	7-21
7.6.2 Comments .....	7-21
7.7 Block Diagonal .....	7-22
7.7.1 Example .....	7-23
7.7.2 Comments .....	7-24
7.8 Concluding Comments .....	7-24
References .....	7-27

## CHAPTER VIII: MULTI-YEAR DYNAMICS AND LINEAR PROGRAMMING

.....	8-1
8.1 Dynamics Background .....	8-1
8.1.1 Should Dynamics be Explicit? .....	8-1
8.1.2 How Long .....	8-2
8.1.3 Time Interval Representation .....	8-3
8.1.4 Initial and Terminal Conditions .....	8-3
8.1.5 Enterprise Life .....	8-4
8.1.6 Time Preference .....	8-4
8.1.7 Risk .....	8-5
8.2 Dynamic Linear Programming .....	8-5
8.2.1 Disequilibrium - Known Life .....	8-5
8.2.1.1 Example .....	8-7
8.2.2 Disequilibrium - Unknown Life .....	8-8
8.2.2.1 Example .....	8-9
8.2.2.2 Comments .....	8-10
8.2.3 Equilibrium - Known Life .....	8-10
8.2.3.1 Example 1 .....	8-11
8.2.3.2 Example 2 .....	8-12
8.2.3.3 Comments .....	8-15
8.2.4 Equilibrium - Unknown Life .....	8-15
8.2.4.1 Example .....	8-16

8.2.5 Overall Comments on Dynamic Linear Programming	8-16
8.3 Recursive Programming	8-17
8.3.1 Example	8-18
8.3.2 Comments on Recursive Programming	8-19
References	8-20

CHAPTER IX: LINEAR PROGRAMMING MODELING:

NONLINEARITIES AND APPROXIMATION	9-1
9.1 Transformations	9-1
9.1.1 Minimization of the Sum of Absolute Deviations	9-1
9.1.1.1 Example	9-3
9.1.1.2 Comments	9-5
9.1.2 Minimization of Largest Absolute Deviation	9-5
9.1.2.1 Example	9-6
9.1.2.2 Comments	9-7
9.1.3 Optimizing a Fraction	9-7
9.1.3.1 Example	9-9
9.1.3.2 Comments	9-10
9.2 Approximations	9-10
9.2.1 Grid Point Approximations	9-10
9.2.1.2 Functions with Separable Variables	9-10
9.2.1.1.1 Example 1.	9-12
9.2.1.1.2 Example 2: Separable Terms in the Constraints	9-13
9.2.1.2 Gridpoints and Gridpoint Refinements	9-14
9.2.1.3 Gridpoint Approximation of Functions of Multiple Variables	9-14
9.2.1.3.1 Homogeneous of Degree 1	9-15
9.2.1.3.2 Homogeneous of Degree Less Than One	9-16
9.2.1.3.3 Comments	9-18
9.2.2 Iterative Approximations	9-19
9.2.3 Other Approximations	9-20
References	9-21

CHAPTER X: MODELING SUMMARY

10.1 Types of Constraints and Variables in Linear Programming Models	10-1
10.1.1 Types of Constraints	10-1
10.1.1.1 Resource Limitations	10-1
10.1.1.2 Minimum Requirements	10-2
10.1.1.3 Supply and Demand Balance	10-2
10.1.1.4 Ratio Control	10-3
10.1.1.5 Bounds	10-4
10.1.1.6 Accounting Relations	10-5
10.1.1.7 Deviation Constraints	10-5
10.1.1.8 Approximation or Convexity Constraints	10-5
10.1.2 Types of Variables	10-5

10.1.2.1 Production Variables . . . . .	10-5
10.1.2.2 Sales Variables . . . . .	10-6
10.1.2.3 Purchase Variables . . . . .	10-6
10.1.2.4 Transformation Variables . . . . .	10-7
10.1.2.5 Slack Variables . . . . .	10-7
10.1.2.6 Surplus Variables . . . . .	10-7
10.1.2.7 Artificial Variables . . . . .	10-7
10.1.2.8 Step Variables . . . . .	10-8
10.1.2.9 Deviation Variables . . . . .	10-8
10.1.2.10 Accounting Variables . . . . .	10-8
10.2 "Violations" of the Algorithmic Assumptions . . . . .	10-8
10.2.1 Nonproportional Example . . . . .	10-9
10.2.2 Non-Additive Example . . . . .	10-10
10.2.3 Uncertainty Examples . . . . .	10-11
10.2.4 Noncontinuous Example . . . . .	10-12
References . . . . .	10-13
CHAPTER XI: MULTI-OBJECTIVE PROGRAMMING . . . . .	11-1
11.1 Formulations . . . . .	11-2
11.1.1 Lexicographic Utility - Target Values . . . . .	11-3
11.1.2 Utility Tradeoff Model . . . . .	11-4
11.1.3 Unknown Utility Function . . . . .	11-6
11.2 Examples . . . . .	11-6
11.2.1 Lexicographic Formulation . . . . .	11-6
11.2.2 Weighted Tradeoff - No Targets . . . . .	11-8
11.2.3 Weighted Objective and Targets . . . . .	11-8
11.3 Choice Among Formulations . . . . .	11-8
11.4 Developing Utility Functions . . . . .	11-9
11.5 Shadow Prices . . . . .	11-11
11.5.1 Example . . . . .	11-12
References . . . . .	11-14
CHAPTER XII: NONLINEAR OPTIMIZATION CONDITIONS . . . . .	12-1
12.1 Optimality Conditions . . . . .	12-1
12.1.1 Unconstrained Optimization . . . . .	12-1
12.1.1.1 Univariate . . . . .	12-1
12.1.1.2 Multivariate functions . . . . .	12-2
12.1.2 Global Optima-Concavity and Convexity . . . . .	12-3
12.1.3 Constrained Optimization . . . . .	12-4
12.1.3.1 Equality Constraints - The Lagrangian . . . . .	12-4
12.1.3.1.1 Second Order Conditions - Constraint Qualifications: Convexity and Concavity . . . . .	12-6
12.1.3.1.2 Interpretation of Lagrange Multipliers . . . . .	12-8
12.1.3.2 Inequality Constraints - Kuhn Tucker Theory . . . . .	12-9
12.1.3.2.1 Example 1 . . . . .	12-11

12.1.3.2.2 Example 2	12-11
12.1.4 Usage of Optimality Conditions	12-12
12.2 Notes on Solution of Nonlinear Programming Models	12-12
12.3 Expressing Nonlinear Programs in Conjunction with GAMS	12-13
References	12-15
CHAPTER XIII: PRICE ENDOGENOUS MODELING	13-1
13.1 Spatial Equilibrium	13-4
13.2 Multi-Market Case	13-8
13.3 Implicit Supply - Multiple Factors/Products	13-9
13.4 Aggregation	13-13
13.5 A Digression on the Assumptions	13-16
13.6 Imperfect Competition	13-18
13.7 Conclusion	13-20
References	13-21
CHAPTER XIV: RISK MODELING	14-1
14.1 Decision Making and Recourse	14-1
14.2 An Aside: Discounting Coefficients	14-2
14.3 Stochastic Programming without Recourse	14-2
14.3.1 Objective Function Coefficient Risk	14-3
14.3.1.1 Mean-Variance Analysis	14-3
14.3.1.1.1 Example	14-4
14.3.1.1.2 Markowitz's E-V Formulation	14-5
14.3.1.1.3 Formulation Choice	14-6
14.3.1.1.4 Characteristics of E-V Model Optimal Solutions	14-6
14.3.1.1.5 E-V Model Use - Theoretical Concerns	14-7
14.3.1.1.6 Specification of the Risk Aversion Parameter	14-7
14.3.1.2 A Linear Approximation - MOTAD	14-8
14.3.1.2.1 Example	14-10
14.3.1.2.2 Comments on MOTAD	14-11
14.3.1.3 Toward A Unified Model	14-13
14.3.1.4 Safety First	14-14
14.3.1.4.1 Example	14-14
14.3.1.4.2 Comments	14-14
14.3.1.5 Target MOTAD	14-15
14.3.1.5.1 Example	14-15
14.3.1.5.2 Comments	14-15
14.3.1.6 DEMP	14-16
14.3.1.6.1 Example	14-16
14.3.1.6.2 Comments	14-17
14.3.1.7 Other Formulations	14-17
14.3.2 Right Hand Side Risk	14-18
14.3.2.1 Chance Constrained Programming	14-18



14.3.2.1.1 Example	14-19
14.3.2.1.2 Comments	14-19
14.3.2.2 A Quadratic Programming Approach	14-20
14.3.3 Technical Coefficient Risk	14-21
14.3.3.1 Merrill's Approach	14-21
14.3.3.2 Wicks and Guise Approach	14-21
14.3.3.2.1 Example	14-22
14.3.3.2.2 Comments	14-23
14.3.4 Multiple Sources of Risk	14-23
14.4 Sequential Risk-Stochastic Programming with Recourse	14-23
14.4.1 Two stage SPR formulation	14-24
14.4.1.1 Example	14-26
14.4.2 Incorporating Risk Aversion	14-27
14.4.2.1 Example	14-28
14.4.3 Extending to Multiple Stages	14-29
14.4.4 Model Discussion	14-30
14.5 General Comments On Modeling Uncertainty	14-30
References	14-33

CHAPTER XV: APPLIED INTEGER PROGRAMMING	15-1
15.1 Why Integer Programming	15-1
15.1.1 Fixed Cost	15-1
15.1.2 Logical Conditions	15-5
15.1.2.1 Either-or-Active Constraints	15-6
15.1.2.2 An Aside: Mutual Exclusivity	15-6
15.1.2.3 Multiple Active Constraints	15-7
15.1.2.4 Conditional Restrictions	15-7
15.1.3 Discrete Levels of Resources	15-8
15.1.4 Distinct Variable Values	15-8
15.1.5 Nonlinear Representations	15-9
15.1.6 Approximation of Nonlinear Functions	15-9
15.2 Feasible Region Characteristics and Solution Difficulties	15-10
15.2.1 Extension to Mixed Integer Feasible Regions	15-12
15.3 Sensitivity Analysis and Integer Programming	15-12
15.4 Solution Approaches to Integer Programming Problems	15-13
15.4.1 Rounding	15-14
15.4.2 Cutting Planes	15-15
15.4.3 Branch and Bound	15-16
15.4.5 Lagrangian Relaxation	15-17
15.4.6 Benders Decomposition	15-18
15.4.7 Heuristics	15-20
15.4.8 Structural Exploitation	15-21
15.4.9 Other Solution Algorithms and Computer Algorithms	15-21
15.5 The Quest for Global Optimality: Non-Convexity	15-21
15.6 Formulation Tricks for Integer Programming - Add More Constraints	15-22

15.7 IP Solutions and GAMS .....	15-23
References .....	15-24
<b>CHAPTER XVI: INTEGER PROGRAMMING FORMULATIONS .....</b>	<b>16-1</b>
16.1 Knapsack - Capital Budgeting Problem .....	16-1
16.1.1 Example .....	16-1
16.1.2 Comments .....	16-2
16.2 Warehouse Location .....	16-2
16.2.1 Example .....	16-4
16.2.2 Comments .....	16-4
16.3 Traveling Salesman Problem .....	16-5
16.3.1 Example .....	16-6
16.3.2 Comments .....	16-7
16.4 Decreasing Costs .....	16-7
16.4.1 Example .....	16-8
16.4.2 Comments .....	16-9
16.5 Machinery Selection .....	16-10
16.5.1 Example .....	16-10
16.5.2 Comments .....	16-11
16.6 Other Formulations .....	16-11
References .....	16-12
<b>CHAPTER XVII: FIXING IMPROPERLY WORKING MODELS .....</b>	<b>17-1</b>
17.1 Unacceptable Solution Conditions .....	17-1
17.1.1 Solver Failure -- Causes and Prevention .....	17-1
17.1.2 Unbounded or Infeasible Solutions .....	17-1
17.1.3 Unsatisfactory Optimal Solutions .....	17-2
17.2 Techniques for Diagnosing Improper Models .....	17-2
17.2.1 Simple Structural Checking .....	17-2
17.2.1.1 Analytical Checking .....	17-2
17.2.1.2 Numerical Model Analysis .....	17-3
17.2.2 A Priori Degeneracy Resolution .....	17-4
17.2.3 Altering Units of Constraints and Variables: Scaling .....	17-4
17.2.3.1 Scaling-The Basic Procedure .....	17-4
17.2.3.2 Mathematical Investigation of Scaling .....	17-7
17.2.3.2.1 Variable Scaling .....	17-7
17.2.3.2.2 Effects of Constraint Scaling .....	17-10
17.2.3.2.3 Objective Function and Right Hand Side Scaling .....	17-11
17.2.3.4 Summary .....	17-12
17.2.3.5 Empirical Example of Scaling .....	17-13
17.2.4 The Use of Artificial Variables to Diagnose Infeasibility .....	17-15
17.2.5 Use Unrealistically Large Upper Bounds to Find Causes of Unboundedness .....	17-17
17.2.6 Budgeting .....	17-18

17.2.7 Row Summing .....	17-21
References .....	17-23
<b>CHAPTER XVIII: VALIDATION OF PROGRAMMING MODELS .....</b>	<b>18-1</b>
18.1 Background .....	18-1
18.2 General Approaches to Validation .....	18-1
18.3 Validation by Construct .....	18-2
18.4 Validation by Results .....	18-3
18.4.1 Parameter Outcome Sets .....	18-3
18.4.2 Validation Experiments .....	18-3
18.4.2.1 Feasibility Experiment .....	18-3
18.4.2.2 Quantity Experiment .....	18-4
18.4.2.3 Price Experiment .....	18-5
18.4.2.4 Prediction Experiment .....	18-5
18.4.2.5 Change Experiment .....	18-5
18.4.2.6 Tracking Experiment .....	18-6
18.4.2.7 Partial Tests .....	18-6
18.4.3 Employing a Validation Test .....	18-6
18.4.4 Evaluation Criteria for Validation .....	18-7
18.5 What if the Model Does not Validate .....	18-8
18.6 Comments .....	18-9
References .....	18-10
<b>CHAPTER XIX: COMPUTERIZED AIDS FOR PROGRAMMING STUDIES .....</b>	<b>19-1</b>
19.1 Model Generation Tools .....	19-1
19.1.1 Potential Functions of a Matrix Generator .....	19-1
19.1.1.1 Input Preparation .....	19-1
19.1.1.2 Formulation Manipulation .....	19-2
19.1.1.3 Documentation .....	19-2
19.1.1.4 Analysis Facilitation .....	19-2
19.1.2 When not to use a Matrix Generator .....	19-2
19.2 Pre-Solution Formulation Analysis Tools .....	19-2
19.3 Solution Interpretation Aids .....	19-3
19.3.1 Solution Summarization .....	19-3
19.3.2 Solution Debugging .....	19-3
19.3.3 Solution Interpretation .....	19-4
19.4 Getting the most from GAMS .....	19-4
19.4.1 Setting Up Data .....	19-4
19.4.2 Changing Model Structure .....	19-4
19.4.3 Providing an Advanced Basis .....	19-5
19.4.4 Report Writing .....	19-5
19.4.5 Debugging Models .....	19-7
19.4.6 Conducting a Comparative Model Analysis .....	19-8
19.4.7 Sensitivity Analysis .....	19-9
19.4.8 Speeding up GAMS .....	19-9

19.4.9 Minimizing Model Size .....	19-9
19.4.10 Avoiding GAMS Failures .....	19-10
References .....	19-11
Appendix I: Using Summation Notation With GAMS .....	AI-1
AI.1 Summation Mechanics .....	AI-1
AI.1.1 Sum of an Item. ....	AI-1
AI.1.2 Multiple Sums .....	AI-1
AI.1.3 Sum of Two Items .....	AI-2
AI.2 Summation Notation Rules .....	AI-2
AI.2.1 For a Scalar Equation .....	AI-2
AI.2.2. For a Family of Equations .....	AI-3
AI.4 Defining and Using Variables .....	AI-5
AI.5 Equations .....	AI-6
AI.6 Cautions and Extensions .....	AI-7
APPENDIX II: EXECUTING GAMS .....	Appendix II-1

**APPLIED MATHEMATICAL PROGRAMMING  
USING ALGEBRAIC SYSTEMS**

by

**Bruce A. McCarl**

**Thomas H. Spreen**

**PREFACE**

**1: INTRODUCTION**

**PART 1. THE BASIC MATHEMATICAL PROBLEM -- LINEAR  
PROGRAMMING**

**2: LINEAR PROGRAMMING**

**3: SOLUTION OF LP PROBLEMS: A MATRIX ALGEBRA  
APPROACH**

**4: DUALITY IN LINEAR PROGRAMMING**

**5: LINEAR PROGRAMMING MODELING**

**6: TOWARD PROPER MODELING**

**7: MORE LINEAR PROGRAMMING MODELING**

**8: MULTI-YEAR DYNAMICS AND LINEAR PROGRAMMING**

**9: LINEAR PROGRAMMING MODELING: NONLINEARITIES  
AND APPROXIMATION**

**10: MODELING SUMMARY**

**PART 2. RELAXATION OF LINEAR PROGRAMMING ASSUMPTIONS**

**11: MULTI-OBJECTIVE PROGRAMMING**

**12: NONLINEAR OPTIMIZATION CONDITIONS**

**13: PRICE ENDOGENOUS MODELING**

**14: RISK MODELING**

**15: APPLIED INTEGER PROGRAMMING**

**16: INTEGER PROGRAMMING FORMULATION**

**PART 3. USE OF MATHEMATICAL PROGRAMMING MODELS**

**17: FIXING IMPROPERLY WORKING MODELS**

**18: VALIDATION OF PROGRAMMING MODELS**

**19: COMPUTERIZED AIDS FOR PROGRAMMING STUDIES**

**APPENDIX 1: USING SUMMATION NOTATION**

**APPENDIX 2: EXECUTING GAMS**

# APPLIED MATHEMATICAL PROGRAMMING USING ALGEBRAIC SYSTEMS

## Preface

This is yet one more book on mathematical programming in use; thus it is useful to ask: Why should anyone be interested?

Mathematical programming is an area in which two separate disciplines have arisen. First there is the algorithm and mathematical properties discipline. People working in this area are interested in the theoretical and computational properties of mathematical programming solution techniques. The majority of mathematical programming related articles, professional journals, and textbooks could be classed into this discipline. The second discipline is applied mathematical programmers. People working in this area are users of the products developed by those in the first area. Their fundamental concerns involve problem formulation, computer specification, solution interpretation and communication with decision makers. Their algorithmic and computational concerns frequently reduce to whether the problem can be solved so that an application can be carried out. Few books address these topics.

This book is aimed toward the second group. Solution methods are discussed only to provide insight into solution interpretation and to a limited extent insight into solution processes.

In that setting we feel the book possesses several features which are notable.

1. An orientation toward applied modelers with solution principles covered only as: a) an aid to problem solution interpretation (Chapters 3 and 12); and b) an aid in formulating problems (Chapter 15).
2. An orientation toward use of the GAMS modeling system (Chapters 5, 12, 15 and 19). Material will be presented on GAMS usage from the users perspective which complements the currently available reference manuals.
3. An orientation toward algebraic modeling including material on algebraic modeling

techniques and a discussion of advantages (Chapter 5 in particular and all subsequent modeling chapters, Appendix 1).

4. An underlying philosophy that mathematical programming modelers need to address all aspects of the problem from data to answer. Thus, procedures for data collection, data computation, problem formulation, model interpretation, and report creation are discussed throughout the book.
5. Presentation of sufficient theory to allow a complete course with the applied work related to the theory.
6. A treatment that spans the Agricultural Economics and Operations Research fields while being accessible to upper division undergraduates and graduate students.
7. Discussion oriented around the assumptions of linear programming and their relaxation (Chapter 1 and Part 2).
8. A detailed discussion of linear programming duality, including a discussion of how duality concepts are useful in modeling and interpretation (Chapters 4 and 5).
9. A modeling discussion which concentrates on models which have been used in application (with associated references) and/or have great application appeal. This discussion is supported by examples which are realistic in terms of problem types and sizes.
10. Unified coverage of topics either not discussed in other books or which are not comprehensively, simultaneously covered elsewhere. Such topics include discussions of:
  - typical linear programming models and their use as building blocks in applications
  - multi year dynamics in linear programming
  - linear programming approaches to non linear programming problems
  - multi-objective programming
  - price endogenous (sector) models



risk

integer programming modeling

model debugging

model validation

extensive and current bibliographic citations

11. Unification of the whole treatment with GAMS, including a disk of the book examples.

## **CHAPTER I: INTRODUCTION**

This book is intended to both serve as a reference guide and a text for a course on Applied Mathematical Programming. The material presented will concentrate upon conceptual issues, problem formulation, computerized problem solution, and results interpretation. Solution algorithms will be treated only to the extent necessary to interpret solutions and overview events which may occur during the solution process.

### **1.1 Practical Problem Analysis**

Problem analysis is by nature an interactive process in which an analyst perceives (or is told about) a problem; conceptualizes an approach; tries out the approach; revises the approach to better fit the problem (alternatively terminates the investigation or tries a new approach) implements the approach; interprets the results; and terminates the inquiry, or transfers the approach to operational personnel. This book will explicitly or implicitly deal with these topics under the assumptions that the problem analysis technique is mathematical programming.

Mathematical programming problem analysis generally have comparative advantage in knowledge of the problem, not in algorithm development procedures. Consequently, the problem analyst should be thoroughly informed on the topics of problem formulation, results interpretation, and model use but in large part can treat the solution processes as a "black box."

### **1.2 Mathematical Programming Approach**

Mathematical programming refers to a set of procedures dealing with the analysis of optimization problems. Optimization problems are generally those in which a decision maker wishes to optimize some measure(s) of satisfaction by selecting values for a set of variables. We will discuss that set of mathematical programs where the variable values are constrained by conditions external to the problem at

hand (for example, constraints on the maximum amount of resources available and/or the minimum amount of certain items which need to be on hand) and sign restrictions on the variables. The general mathematical programming problem we will treat is:

$$\begin{array}{ll} \text{Optimize} & F(X) \\ \text{Subject To (s.t.)} & G(X) \in S_1 \\ & X \in S_2 \end{array}$$

Here  $X$  is a vector of decision variables. The level of  $X$  is chosen so that an objective is optimized where the objective is expressed algebraically as  $F(X)$  which is called the objective function. This objective function will be maximized or minimized. However, in setting  $X$ , a set of constraints must be obeyed requiring that functions of the  $X$ 's behave in some manner. These constraints are reflected algebraically by the requirements that: a)  $G(X)$  must belong to  $S_1$  and b) the variables individually must fall into  $S_2$ .

The mathematical programming problem encompasses many different types of problems some of which will be discussed in this book. In particular, if  $F(X)$  and  $G(X)$  are linear and the  $X$ 's are individually non-negative, then the problem becomes a linear programming problem. If the  $X \in S_2$  restriction requires some  $X$ 's to take on integer values, then this is an integer programming problem. If  $G(X)$  is linear,  $F(X)$  quadratic, and the  $S_2$  restrictions are simply non-negativity restrictions, then we have a quadratic programming problem. Finally, if  $F(X)$  and  $G(X)$  are general nonlinear functions with  $S_2$  being nonnegativity conditions, the problem is a nonlinear programming problem.

### 1.3 Mathematical Programming in Use

Mathematical programming is most often thought of as a technique which decision makers can use to develop optimal values of the decision variables. However, there are a considerable number of other potential usages of mathematical programming. Furthermore, as we will argue below, numerical usage for

identification of specific decisions is probably the least common usage in terms of relative frequency.

Three sets of usages of mathematical programming that we regard as common are: 1) problem insight construction; 2) numerical usages which involve finding model solutions; and 3) solution algorithm development and investigation. We will discuss each of these in turn.

### **1.3.1 Generating Problem Insight**

Mathematical programming forces one to state a problem carefully. One must define: a) decision variables; b) constraints; c) the objective function; d) linkages between variables and constraints that reflects complementary, supplementary and competitive relationships among variables; and e) consistent data. The decision maker is forced to understand the problem interacting with the situation thoroughly, discovering relevant decision variables and constraining factors. Frequently, the resultant knowledge outweighs the value of any solutions and is probably the number one benefit of most mathematical programming exercises.

A second insight generating usage of mathematical programming involves analytical investigation of problems. While it is not generally acknowledged that mathematical programming is used, it provides the underlying basis for a large body of microeconomic theory. Often one sets up, for example, a utility function to be maximized subject to a budget constraint, then uses mathematical programming results for the characterization of optimal values. In turn, it is common to derive theoretical conclusions and state the assumptions under which those conclusions are valid. This is probably the second most common usage of mathematical programming and again is a nonnumerical use.

### **1.3.2 Numerical Mathematical Programming**

Numerical usages fall into four subclasses: 1) prescription of solutions; 2) prediction of consequences; 3) demonstration of sensitivity; and 4) solution of systems of equations. The most commonly thought of application of mathematical programming involves the prescriptive or normative

question: Exactly what decision should be made given a particular specification of objectives, variables, and constraints? This is most often perceived as the usage of mathematical programming, but is probably the least common usage over the universe of models. In order to understand this assertion, one simply has to address the question: "Do you think that many decision makers yield decision making power to a model?" Very few circumstances allow this kind of trust. Most often, models are used for decision guidance or to predict the consequences of actions. One should adopt the philosophical position that models are an abstraction of reality and that an abstraction will yield a solution suggesting a practical solution, not always one that should be implemented.

The second numerical mathematical programming usage involves prediction. Here the model is assumed to be an adequate depiction of the entity being represented and is used to predict in a conditional normative setting. Typically, this occurs in a business setting where the model is used to predict the consequences of environmental alterations (caused by investments, acquisition of resources, weather changes, market price conditions, etc.). Similarly, models are commonly used in government policy settings to predict the consequences of policy changes. Models have been used, for example, to analyze the implications for social benefits of a change in ambient air quality. Predictive use is probably the most common numerical usage of mathematical programming.

The third and next most common numerical usage of mathematical programming is sensitivity demonstration. Many Ph.D. theses are done where no one ever tries to implement the solutions, and no one ever uses the solutions for predictions. Rather, the model is used to demonstrate what might happen if certain factors are changed. Here the model is usually specified with a "realistic" data set, then is used to demonstrate the implications of alternative input parameter and constraint specifications.

The final numerical use is as a technical device in empirical problems. Mathematical programs can be used to develop such things as solutions to large systems of equations, equation fits which minimize

absolute deviations, or equation fits which result in all positive or all negative error terms. In this case, the ability of modern day solvers to treat problems with thousands of variables and constraints may be called to use. For example, a large USDA econometric model was solved for a time using a mathematical programming solver.

### **1.3.3 Algorithmic Development**

Much of the mathematical programming related effort involves solution algorithm development. Formally, this is not a usage, but an enormous amount of work is done here as is evidenced by the many textbooks treating this topic. In such a setting the mathematical programming model is used as a vehicle for solution technique development. Work is also done on new formulation techniques and their ability to appropriately capture applied problems.

## **1.4 Book Plan**

Mathematical programming in application consists, to a large degree, of applied linear programming. This book will not neglect that. Chapters II-X will cover linear solution procedures, duality, modeling, and advanced modeling, computational issues. Discussion will then move onto nonlinear programming covering the general case, then price endogenous programming, risk programs, and integer programming.

## CHAPTER II: LINEAR PROGRAMMING

The most fundamental optimization problem treated in this book is the linear programming (LP) problem. In the LP problem, decision variables are chosen so that a linear function of the decision variables is optimized and a simultaneous set of linear constraints involving the decision variables is satisfied.

### 2.1 The Basic LP Problem

An LP problem contains several essential elements. First, there are decision variables( $x_j$ ) the level of which denotes the amount undertaken of the respective unknowns of which there are  $n$  ( $j=1, 2, \dots, n$ ). Next is the linear objective function where the total objective value ( $Z$ ) equals  $c_1x_1 + c_2x_2 + \dots + c_nx_n$ . Here  $c_j$  is the contribution of each unit of  $x_j$  to the objective function. The problem is also subject to constraints of which there are  $m$ . An algebraic expression for the  $i^{\text{th}}$  constraint is  $a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i$  ( $i=1, 2, \dots, m$ ) where  $b_i$  denotes the upper limit or right hand side imposed by the constraint and  $a_{ij}$  is the use of the items in the  $i^{\text{th}}$  constraint by one unit of  $x_j$ . The  $c_j$ ,  $b_i$ , and  $a_{ij}$  are the data (exogenous parameters) of the LP model.

Given these definitions, the LP problem is to choose  $x_1, x_2, \dots, x_n$  so as to

$$\begin{aligned}
\text{Maximize (Max) } Z &= c_1x_1 + c_2x_2 + \dots + c_nx_n \\
\text{subject to (s.t.)} & a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1 \\
& a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2 \\
& \cdot \\
& \cdot \\
& a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m \\
& x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0
\end{aligned}$$

This formulation may also be expressed in matrix notation.

$$\begin{aligned}
\text{Max } Z &= CX \\
\text{s.t.} & AX \leq b \\
& X \geq 0 \\
\text{where } C_{1 \times n} &= [c_1, c_2, \dots, c_n]
\end{aligned}$$

$$X_{n \times 1} = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{bmatrix} \quad b_{m \times 1} = \begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ b_m \end{bmatrix} \quad A_{m \times n} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

Many variants have been posed of the above problem and applications span a wide variety of settings. For example, the basic problem could involve setting up: a) a livestock diet determining how



much of each feed stuff to buy so that total diet cost is minimized subject to constraints on minimum and maximum levels of nutrients; b) a production plan where the firm chooses the profit maximizing level of production subject to resource (labor and raw materials) constraints; or c) a minimum cost transportation plan determining the amount of goods to transport across each available route subject to constraints on supply availability and demand.

## 2.2 Basic LP Example

For further exposition of the LP problem it is convenient to use an example. Consequently, consider the decision problem of Joe's van conversion shop. Suppose Joe makes custom vans and can produce either fine or fancy vans. The decision modeled is how many of each van type to convert this week. The number converted this week by van type constitutes the decision variables. We denote these variables as  $x_{\text{fine}}$  and  $x_{\text{fancy}}$ . Now suppose Joe can sell fancy vans for \$13,000, but to do this he has to buy a \$10,000 plain van and use \$1,000 worth of materials leaving a \$2,000 profit margin. Similarly, the profit margin on fine vans is \$1,700. Joe figures the shop can work on no more than 12 vans in a week. Joe hires 7 people including himself and operates 8 hours per day, 5 days a week and thus has at most 280 hours of labor available in a week. Joe also estimates that a fancy van will take 25 hours of labor, while a fine van will take 20 hours.

In order to set up Joe's problem as an LP, we must mathematically express the objective and constraint functions. Since the estimated profit per fancy vans is \$2,000 per van, then  $2,000x_{\text{fancy}}$  is the total profit from all the fancy vans produced. Similarly,  $1,700x_{\text{fine}}$  is the total profit from fine van production. The total profit from all van conversions is  $2,000x_{\text{fancy}} + 1,700x_{\text{fine}}$ . This equation mathematically describes the total profit consequences of Joe's choice of the decision variables. Given that Joe wishes to maximize total profit, his objective is to determine the levels of the decision variables which

will

$$\text{Max } Z = 2,000x_{\text{fancy}} + 1,700x_{\text{fine}}$$

This is the objective function of the LP model.

Joe's factory has limited amounts of capacity and labor. In this case, capacity and labor are resources which limit the allowable (also called feasible) values of the decision variables. Since the decision variables are defined in terms of vans converted in a week, the total vans converted is  $x_{\text{fancy}} + x_{\text{fine}}$ . This sum must be less than or equal to the capacity available (12). Similarly, total labor use is given by  $25x_{\text{fancy}} + 20x_{\text{fine}}$  which must be less than or equal to the labor available (280). These two limits are called constraints. Finally, it makes no sense to convert a negative number of vans of either type; thus,  $x_{\text{fancy}}$  and  $x_{\text{fine}}$  are restricted to be greater than or equal to zero. Putting it all together, the LP model of Joe's problem is to choose the values of  $x_{\text{fancy}}$  and  $x_{\text{fine}}$  so as to:

$$\begin{array}{llll} \text{Max } Z = & 2,000x_{\text{fancy}} & + & 1,700x_{\text{fine}} \\ \text{s.t.} & x_{\text{fancy}} & + & x_{\text{fine}} \leq 12 \\ & 25x_{\text{fancy}} & + & 20x_{\text{fine}} \leq 280 \\ & x_{\text{fancy}} & , & x_{\text{fine}} \geq 0 \end{array}$$

This is a formulation of Joe's LP problem depicting the decision to be made (i.e. the choice of  $x_{\text{fancy}}$  and  $x_{\text{fine}}$ ). The formulation also identifies the rules, commonly called constraints, by which the decision is made and the objective which is pursued in setting the decision variables.

### 2.3 Other Forms of the LP Problem

Not all LP problems will naturally correspond to the above form. Other legitimate representations of LP models are:

- 1) Objectives which involves minimize instead of maximize i.e.,

$$\text{Minimize } Z = c_1x_1 + c_2x_2 + \dots + c_nx_n.$$

- 2) Constraints which are "greater than or equal to" instead of "less than or equal to"; i.e.,

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \geq b_1.$$

- 3) Constraints which are strict equalities; i.e.,

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1.$$

- 4) Variables without non-negativity restriction i.e.,  $x_j$  can be unrestricted in sign i.e.,  $x_j \begin{matrix} < \\ > \end{matrix} 0$ .

- 5) Variables required to be non-positive i.e.,  $x_j \leq 0$ .

## 2.4 Assumptions of LP

LP problems embody seven important assumptions relative to the problem being modeled. The first three deal with the appropriateness of the formulation; the last four deal with the mathematical relationships within the model.

### 2.4.1 Objective Function Appropriateness

This assumption means that within the formulation the objective function is the sole criteria for choosing among the feasible values of the decision variables. Satisfaction of this assumption can often be difficult as, for example, Joe might base his van conversion plan not only on profit but also on risk exposure, availability of vacation time, etc. The risk modeling and multi-objective chapters cover the relaxation of this assumption.

### 2.4.2 Decision Variable Appropriateness

A key assumption is that the specification of the decision variables is appropriate. This assumption requires that

- a) The decision variables are all fully manipulatable within the feasible region and are under

the control of the decision maker.

- b) All appropriate decision variables have been included in the model.

The nature and relaxation of sub-assumption (a) is discussed in the Advanced modeling considerations chapter in the "Common Mistakes" section, as is sub-assumption (b). Sub-assumption © is also highlighted in Chapters IX and XVI.

### **2.4.3 Constraint Appropriateness**

The third appropriateness assumption involves the constraints. Again, this is best expressed by identifying sub-assumptions:

- a) The constraints fully identify the bounds placed on the decision variables by resource availability, technology, the external environment, etc. Thus, any choice of the decision variables which simultaneously satisfies all the constraints is admissible.
- b) The resources used and/or supplied within any single constraint are homogeneous items which can be used or supplied by any decision variable appearing in that constraint.
- c) Constraints have not been imposed which improperly eliminate admissible values of the decision variables.
- d) The constraints are inviolate. No considerations involving model variables other than those included in the model can lead to the relaxation of the constraints.

Relaxations and/or the implications of violating these assumptions are discussed throughout the text.

### **2.4.4 Proportionality**

Variables in LP models are assumed to exhibit proportionality. Proportionality deals with the contribution per unit of each decision variable to the objective function. This contribution is assumed constant and independent of the variable level. Similarly, the use of each resource per unit of each decision

variable is assumed constant and independent of variable level. There are no economies of scale.

For example, in the general LP problem, the net return per unit of  $x_j$  produced is  $c_j$ . If the solution uses one unit of  $x_j$ , then  $c_j$  units of return are earned, and if 100 units are produced, then returns are  $100c_j$ . Under this assumption, the total contribution of  $x_j$  to the objective function is always proportional to its level.

This assumption also applies to resource usage within the constraints. Joe's labor requirement for fine vans was 25 hours/van. If Joe converts one fine van he uses 25 hours of labor. If he converts 10 fine vans he uses 250 hours ( $25 \cdot 10$ ). Total labor use from van conversion is always strictly proportional to the level of vans produced.

Economists encounter several types of problems in which the proportionality assumption is grossly violated. In some contexts, product price depends upon the level of production. Thus, the contribution per unit of an activity varies with the level of the activity. Methods to relax the proportionality assumption are discussed in the nonlinear approximations, price endogenous, and risk chapters. Another case occurs when fixed costs are to be modeled. Suppose there is a fixed cost associated with a variable having any non-zero value (i.e., a construction cost). In this case, total cost per unit of production is not constant. The integer programming chapter discusses relaxation of this assumption.

#### **2.4.5 Additivity**

Additivity deals with the relationships among the decision variables. Simply put their contributions to an equation must be additive. The total value of the objective function equals the sum of the contributions of each variable to the objective function. Similarly, total resource use is the sum of the resource use of each variable. This requirement rules out the possibility that interaction or multiplicative terms appear in the objective function or the constraints.

For example, in Joe's van problem, the value of the objective function is 2,000 times the fancy vans

converted plus 1,700 times the fine vans converted. Converting fancy vans does not alter the per van net margin of fine vans and vice versa. Similarly, total labor use is the sum of the hours of labor required to convert fancy vans and the hours of labor used to convert fine vans. Making a lot of one van does not alter the labor requirement for making the other.

In the general LP formulation, when considering variables  $x_j$  and  $x_k$ , the value of the objective function must always equal  $c_j$  times  $x_j$  plus  $c_k$  times  $x_k$ . Using  $x_j$  does not affect the per unit net return of  $x_k$  and vice versa. Similarly, total resource use of resource I is the sum of  $a_{ij}x_j$  and  $a_{ik}x_k$ . Using  $x_j$  does not alter the resource requirement of  $x_k$ . The nonlinear approximation, price endogenous and risk chapters present methods of relaxing this assumption.

#### **2.4.6 Divisibility**

The problem formulation assumes that all decision variables can take on any non-negative value including fractional ones; (i.e., the decision variables are continuous). In the Joe's van shop example, this means that fractional vans can be converted; e.g., Joe could convert 11.2 fancy vans and 0.8 fine vans.

This assumption is violated when non-integer values of certain decision variables make little sense. A decision variable may correspond to the purchase of a tractor or the construction of a building where it is clear that the variable must take on integer values. In this case, it is appropriate to use integer programming.

#### **2.4.7 Certainty**

The certainty assumption requires that the parameters  $c_j$ ,  $b_i$ , and  $a_{ij}$  be known constants. The optimum solution derived is predicated on perfect knowledge of all the parameter values. Since all exogenous factors are assumed to be known and fixed, LP models are sometimes called non-stochastic as contrasted with models explicitly dealing with stochastic factors. This assumption gives rise to the term "deterministic" analysis.

The exogenous parameters of a LP model are not usually known with certainty. In fact, they are usually estimated by statistical techniques. Thus, after developing a LP model, it is often useful to conduct sensitivity analysis by varying one of the exogenous parameters and observing the sensitivity of the optimal solution to that variation. For example, in the van shop problem the net return per fancy van is \$2,000, but this value depends upon the van cost, the cost of materials and the sale price all of which could be random variables.

Considerable research has been directed toward incorporating uncertainty into programming models. We devote a chapter to that topic.

## **CHAPTER III SOLUTION OF LP PROBLEMS: A MATRIX ALGEBRA**

### **APPROACH**

Linear programming solution has been the subject of many articles and books. Complete coverage of LP solution approaches is beyond the scope of this book and is present in many other books. However, an understanding of the basic LP solution approach and the resulting properties are of fundamental importance. Thus, we cover LP solution principles from a matrix algebra perspective demonstrating the simplex algorithm and the properties of optimal solutions. In addition, we cover several practical matters.

### 3.1 Matrix Formulation of the Linear Programming Problem

The matrix version of the basic LP problem can be expressed as in the equations below.

$$\begin{aligned} \text{Max } & CX \\ \text{s.t. } & AX \leq b \\ & X \geq 0 \end{aligned}$$

Here the term  $CX$  is maximized where  $C$  is an  $1 \times N$  vector of profit contributions and  $X$  is an  $N \times 1$  vector of decision variables. This maximization is subject to inequality constraints involving  $M$  resources so that  $A$  is an  $M \times N$  matrix giving resource use coefficients by the  $X$ 's, and  $b$  is an  $M \times 1$  vector of right hand side or resource endowments. We further constrain  $X$  to be non-negative in all elements.

It is common to convert the LP inequality system to equalities by adding slack variables. These variables account for the difference between the resource endowment ( $b$ ) and the use of resources by the variables ( $AX$ ) at no cost to the objective function. Thus, define

$$S = b - AX$$

as the vector of slack variables. Each slack variable is restricted to be nonnegative thereby insuring that resource use is always less than or equal to the resource endowment. One slack variable is added for each constraint equation. Rewriting the constraints gives

$$AX + IS = b,$$

where  $I$  is an  $M \times M$  identity matrix and  $S$  is a  $M \times 1$  vector. Also the slack variables appear in the objective function with zero coefficients. Thus, we add an  $1 \times M$  vector of zero's to the objective function and conditions constraining the slack variables to be nonnegative. The resultant augmented LP is

$$\begin{aligned} \text{Max } & CX \quad 0 \quad OS \\ \text{s.t. } & AX \quad 0 \quad IS \leq b \\ & X, \quad S \geq 0. \end{aligned}$$



Throughout the rest of this section we redefine the X vector to contain both the original X's and the slacks. Similarly, the new C vector will contain the original C along with the zeros for the slacks, and the new A matrix will contain the original A matrix along with the identity matrix for the slacks. The resultant problem is

$$\begin{aligned} \text{Max } & CX \\ \text{s.t. } & AX \leq b \\ & X \geq 0 \end{aligned}$$

### 3.2 Solving LP's by Matrix Algebra

LP theory (Dantzig(1963); Bazarra, et al.) reveals that a solution to the LP problem will have a set of potentially nonzero variables equal in number to the number of constraints. Such a solution is called a **Basic Solution** and the associated variables are commonly called **Basic Variables**. The other variables are set to zero and are called the **nonbasic variables**. Once the basic variables have been chosen; the X vector may be partitioned into  $X_B$ , denoting the vector of the basic variables, and  $X_{NB}$ , denoting the vector of the nonbasic variables. Subsequently, the problem is partitioned to become

$$\begin{aligned} \text{Max } & C_B X_B + C_{NB} X_{NB} \\ \text{s.t. } & B X_B + A_{NB} X_{NB} \leq b \\ & X_B, X_{NB} \geq 0. \end{aligned}$$

The matrix B is called the **Basis Matrix**, containing the coefficients of the basic variables as they appear in the constraints.  $A_{NB}$  contains the coefficients of the nonbasic variables. Similarly  $C_B$  and  $C_{NB}$  are the objective function coefficients of the basic and nonbasic variables.

Now suppose we address the solution of this problem via the simplex method. The simplex solution approach relies on choosing an initial B matrix, and then interactively making improvements.

Thus, we need to identify how the solution changes when we change the B matrix. First, let us look at how the basic solution variable values change. If we rewrite the constraint equation as

$$BX_B = b - A_{NB} X_{NB}.$$

Setting the nonbasic variables ( $X_{NB}$ ) to zero gives

$$BX_B = b.$$

This equation may be solved by premultiplying both sides by the inverse of the basis matrix (assuming non-singularity) to obtain the solution for the basic variables,

$$B^{-1}BX_B = I X_B = B^{-1}b \quad \text{or} \quad X_B = B^{-1}b.$$

We may also examine what happens when the nonbasic variables are changed from zero. Multiply both sides of the equation including the nonbasic variables by  $B^{-1}$  giving

$$X_B = B^{-1}b - B^{-1}A_{NB} X_{NB}.$$

This expression gives the values of the basic variables in terms of the basic solution and the nonbasic variables. This is one of the two fundamental equations of LP. Writing the second term of the equation in summation form yields

$$X_B = B^{-1}b - \sum_{j, NB} B^{-1}a_j x_j$$

where NB gives the set of nonbasic variables and  $a_j$  the associated column vectors for the nonbasic variables  $x_j$  from the original A matrix. This equation shows how the values of the basic variables are altered as the value of nonbasic variables change. Namely, if all but one ( $x_0$ ) of the nonbasic variables are

left equal to zero then this equation becomes

$$X_B = B^{-1}b - B^{-1}a_0x_0$$

This gives a simultaneous system of equations showing how all of the basic variables are affected by changes in the value of a nonbasic variable. Furthermore, since the basic variables must remain non-negative the solution must satisfy

$$X_B = B^{-1}b - B^{-1}a_0x_0 \geq 0.$$

This equation permits the derivation of a bound on the maximum amount the nonbasic variable  $x_0$  can be changed while the basic variables remain non-negative. Namely,  $x_0$  may increase until one of the basic variables becomes zero. Suppose that the first element of  $X_B$  to become zero is  $x_{Bi^*}$ . Solving for  $x_{Bi^*}$  gives

$$x_{Bi^*} = (B^{-1}b)_{i^*} - (B^{-1}a_0)_{i^*}x_0 = 0$$

where  $(\ )_{i^*}$  denotes the  $i^{\text{th}}$  element of the vector. Solving for  $x_0$  yields

$$x_0 = (B^{-1}b)_{i^*} / (B^{-1}a_0)_{i^*}, \text{ where } (B^{-1}a_0)_{i^*} < 0.$$

This shows the value of  $x_0$  which causes the  $i^{\text{th}}$  basic variable to become zero. Now since  $x_0$  must be nonnegative then we need only consider cases in which a basic variable is decreased by increasing the nonbasic variable. This restricts attention to cases where  $(B^{-1}a_0)_i$  is positive. Thus, to preserve non-negativity of all variables, the maximum value of  $x_0$  is

$$x_0 = \min (B^{-1}b)_i / (B^{-1}a_0)_i \text{ for all } i \text{ where } (B^{-1}a_0)_i > 0$$

The procedure is called the minimum ratio rule of linear programming. Given the identification of a nonbasic variable, this rule gives the maximum value the entering variable can take on. We also know that if  $i^*$  is the row where the minimum is attained then the basic variable in that row will become zero. Consequently, that variable can leave the basis with  $x_0$  inserted in its place. Note, if the minimum ratio rule reveals a tie, (i.e., the same minimum ratio occurs in more than one row), then more than one basic

variable reaches zero at the same time. In turn, one of the rows where the tie exists is arbitrarily chosen as  $I^*$  and the new solution has at least one zero basic variable and is degenerate<sup>1</sup>. Also, note that if all the coefficients of  $x_0$  are zero or negative --  $(B^{-1}a_0)_I \geq 0$  -- for all  $I$  -- then this would indicate an unbounded solution, if increasing the value of the nonbasic variable increases the objective function, since the variable does not decrease the value of any basic variables.

Another question is which nonbasic variable should be increased? Resolution of this question requires consideration of the objective function. The objective function, partitioned between the basic and nonbasic variables, is given by

$$Z = C_B X_B + C_{NB} X_{NB}$$

Substituting the  $X_B$  equation (3.1) yields

$$Z = C_B (B^{-1}b + B^{-1}A_{NB} X_{NB}) + C_{NB} X_{NB}$$

or

$$Z = C_B B^{-1}b + C_B B^{-1}A_{NB} X_{NB} + C_{NB} X_{NB}$$

or

$$Z = C_B B^{-1}b + (C_B B^{-1}A_{NB} + C_{NB}) X_{NB}$$

This is the second fundamental equation of linear programming. Expressing the second term in summation notation yields

$$Z = C_B B^{-1}b + \sum_{j, NB} (C_B B^{-1}a_j + c_j) x_j.$$

This expression gives both the current value of the objective function for the basic solution ( $C_B B^{-1}b$  since all nonbasic  $x_j$  equal zero) and how the objective function changes given a change in the value of nonbasic variables. Namely, when changing  $x_0$

$$Z = C_B B^{-1}b + (C_B B^{-1}a_0 + c_0) x_0.$$

Since the first term of the equation is equal to the value of the current objective function, ( $\bar{Z}$ ), then

---

<sup>1</sup> A degenerate solution is defined to be one where at least one basic variable equals zero.

it can be rewritten as

$$Z = \bar{Z} + (C_B B^{-1} a_0 - c_0) X_0, \dots$$

For maximization problems, the objective value will increase for any entering nonbasic variable if its term,  $C_B B^{-1} a_0 - c_0$ , is negative. The criterion that is most commonly used to determine which variable to enter is: select the nonbasic variable that increases the value of objective function the most per unit of the variable entered. Thus, we choose the variable to enter as that variable  $x_0$  such that the value of  $C_B B^{-1} a_0 - c_0$  is most negative. This is the simplex criterion rule of linear programming and the term  $C_B B^{-1} a_0 - c_0$  is called the reduced cost. If there are no variables with negative values of  $C_B B^{-1} a_0 - c_0$  then the solution cannot be improved on and is optimal. However, if a variable is identified by this rule then it should be entered into the basis. Since the basis always has a number of variables equal to the number of constraints, then to put in a new variable one of the old basic variables must be removed. The variable to remove is that basic variable which becomes zero first as determined by the minimum ratio rule. This criteria guarantees the non-negativity condition is maintained providing the initial basis is non-negative. These results give the fundamental equations behind the most popular method for solving LP problems which is the simplex algorithm. (Karmarkar presents an alternative method which is just coming into use.)

### 3.2.1 The Simplex Algorithm

Formally, the matrix algebra version of the simplex algorithm (assuming that an initial feasible invertible basis has been established) for a maximization problem follows the steps:

- 1) Select an initial feasible basis  $B$ ; commonly this is composed of all slack variables and is the identity matrix.
- 2) Calculate the Basis inverse ( $B^{-1}$ ).
- 3) Calculate  $C_B B^{-1} a_j - c_j$  for the nonbasic variables and identify the entering variable as the variable which yields the most negative value of that calculation; denote that variable as  $x_0$ ; if there are none, go to step 6.

- 4) Calculate the minimum ratio rule.

$$\text{Min}_i (B^{-1}b)_i / (B^{-1}a_0)_i \quad \text{where } (B^{-1}a_0)_i > 0$$

Denote the row where the minimum ratio occurs as row  $I^*$ ; if there are no rows with  $(B^{-1}a_0)_i > 0$  then go to step 7.

- 5) Remove the variable that is basic in row  $I^*$  by replacing the variable in the  $I^*$ th column of the basis matrix with column  $a_0$  and recalculate the basis inverse. Go to step 3.
- 6) The solution is optimal. The optimal variable values equal  $B^{-1}b$  for the basic variables and zero for the nonbasic variables. The optimal reduced costs are  $C_B B^{-1}a_j - c_j$  (also commonly called  $Z_j - c_j$ ). The optimal value of the objective function is  $C_B B^{-1}b$ . Terminate.
- 7) The problem is unbounded. Terminate.

### 3.2.2 Example

Suppose we solve Joe's van conversion problem from Chapter II. After adding slacks that problem becomes

$$\begin{array}{llllll} \text{Max} & 2000X_{\text{fancy}} & \% & 1700X_{\text{fine}} & \% & 0s_1 & \% & 0s_2 \\ \text{s.t.} & X_{\text{fancy}} & \% & X_{\text{fine}} & \% & s_1 & & 12 \\ & 25X_{\text{fancy}} & \% & 20X_{\text{fine}} & & & \% & s_2 & 280 \end{array}$$

Now suppose we choose  $s_1$  and  $s_2$  to be in the initial basis. Thus, initially

$$C = [2000 \ 1700 \ 0 \ 0] \quad b = \begin{bmatrix} 12 \\ 280 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 25 & 20 & 0 & 1 \end{bmatrix} \quad X_{\text{NB}} = \begin{bmatrix} X_{\text{fancy}} \\ X_{\text{fine}} \end{bmatrix}$$

$$C_B = [0 \ 0] \quad C_{NB} = [2000 \ 1700]$$

$$B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad A_{NB} = \begin{bmatrix} 1 & 1 \\ 25 & 20 \end{bmatrix}$$

$$B^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Now using criterion for selecting the entering variables ( $C_B B^{-1} a_{NB} - C_{NB}$ ):

$$C_B B^{-1} A_{NB} - C_{NB} = [-2000 \ 1700]$$

Taking the variable associated with the most negative value (-2000) from this calculation indicates the first nonbasic variable  $X_{fancy}$ , should enter. Computation of the minimum ratio rule requires the associated  $B^{-1}a_1$  and  $B^{-1}b$

$$B^{-1}a_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 25 \end{bmatrix} = \begin{bmatrix} 1 \\ 25 \end{bmatrix}$$

$$X_B = B^{-1}b = \begin{bmatrix} 12 \\ 280 \end{bmatrix}$$

Using the criterion for leaving variable

$$\min_i [(B^{-1}b)_i / (B^{-1}a_1)_i] = \min \left[ \frac{12/1}{280/25} \right] = \left[ \frac{12}{11.2} \right] = 11.2 \quad i = 2$$

In this case, the minimum ratio occurs in row 2. Thus, we replace the second basic variable,  $s_2$ , with  $X_{fancy}$ .

At this point, the new basic and nonbasic items become

$$X_B = \begin{bmatrix} s_1 \\ X_{fancy} \end{bmatrix} \quad X_{NB} = \begin{bmatrix} X_{fine} \\ s_2 \end{bmatrix}$$

$$C_B = [0 \ 2000] \quad C_{NB} = [1700 \ 0]$$

$$B = \begin{bmatrix} 1 & 1 \\ 0 & 25 \end{bmatrix} \quad A_{NB} = \begin{bmatrix} 1 & 0 \\ 20 & 1 \end{bmatrix}$$

and the new basis inverse is

$$B^{-1} = \begin{bmatrix} 1 & 1/25 \\ 0 & 1/25 \end{bmatrix}$$

Recomputing the reduced costs for the nonbasic variables  $X_{fine}$ , and  $s_2$  gives

$$C_B B^{-1} A_{NB} - C_{NB} = [0 \ 2000] \begin{bmatrix} 1 & 1/25 \\ 0 & 1/25 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 20 & 1 \end{bmatrix} - [1700 \ 0] = [100 \ 80]$$

Observe that the procedure implies  $X_{fine}$  should enter this basis. The coefficients for the minimum ratio rule are

$$B^{-1} a_2 = \begin{bmatrix} 1 & 1/25 \\ 0 & 1/25 \end{bmatrix} \begin{bmatrix} 1 \\ 20 \end{bmatrix} = \begin{bmatrix} 1/5 \\ 4/5 \end{bmatrix}, \quad B^{-1} b = \begin{bmatrix} 0.8 \\ 11.2 \end{bmatrix}$$

The minimum ratio rule computation yields

$$\text{Min} \left[ \begin{array}{l} 0.8/(1/5) \\ 11.2/(4/5) \end{array} \right] = \left[ \begin{array}{l} 4 \\ 14 \end{array} \right], \quad 4 < 14 \Rightarrow i = 1$$

In the current basis,  $s_1$  is the basic variable associated with row 1. Thus, replace  $s_1$  with  $X_{fine}$ . The new basis vector is  $[X_{fine} \ X_{fancy}]$  and the basic matrix is now

$$B = \begin{bmatrix} 1 & 1 \\ 20 & 25 \end{bmatrix}$$

In turn the basis inverse becomes

$$B^{-1} = \begin{bmatrix} 5 & 1/5 \\ 4 & 1/5 \end{bmatrix}$$

The resultant reduced costs are



$$C_B B^{-1} A_{NB} \text{ \& } C_{NB} = [1700 \ 2000] \begin{bmatrix} 5 & 1/5 \\ 4 & 1/5 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ \& } [0 \ 0] = [500 \ 60]$$

Since all of these are greater than zero, this solution is optimal. In this optimal solution

$$X_B = \begin{bmatrix} X_{\text{fine}} \\ X_{\text{fancy}} \end{bmatrix}, \quad B^{-1}b = \begin{bmatrix} 5 & 1/5 \\ 4 & 1/5 \end{bmatrix} \begin{bmatrix} 12 \\ 280 \end{bmatrix} = \begin{bmatrix} 4 \\ 8 \end{bmatrix}$$

$$Z = C_B X_B = 22800$$

$$C_B B^{-1} a_j - c_j = [0 \ 0 \ 500 \ 60] \text{ \& } j \in NB$$

This method may be expanded to handle difficulties with finding the initial nonnegative basis using either the Phase I/Phase II or BIG M methods discussed below.

### 3.3 Solutions and Their Interpretation

LP solutions arise and are composed of a number of elements. In this section we discuss general solution interpretation, common solver solution format and contents, special solution cases and sensitivity analysis.

#### 3.3.1 General Solution Interpretation

The two fundamental equations developed in section 3.1 may be utilized to interpret the LP solution information. The first (3.1) shows how the basic variables change as nonbasic variables are changed,

$$X_B = B^{-1}b + \sum_{j \in NB} B^{-1}a_j x_j$$

and the second (3.2) give the associated change in the objective function when a nonbasic variable is changed

$$Z = C_B B^{-1}b + \sum_{j \in NB} (C_B B^{-1}a_j - c_j) x_j$$

Suppose we assume that an optimal basic solution has been found and that  $B$  and  $B^{-1}$  are the associated basis and basis inverse. Now suppose we consider changing the constraint right hand sides. The implications of

such a change for the solution information may be explored using calculus. Differentiating the above equations with respect to the right hand side  $b$  yields

$$\frac{\partial Z}{\partial b} = C_B B^{-1}$$

$$\frac{\partial X_B}{\partial b} = B^{-1}$$

These results indicate that  $C_B B^{-1}$  is the expected rate of change in the objective function when the right hand sides are changed. The values  $C_B B^{-1}$  are called the shadow prices and give estimates of the marginal values of the resources (later they will also be called the Dual Variables or Dual Solution). Similarly,  $B^{-1}$  gives the expected rate of change in the basic variables when resources are changed. Thus when the first right hand side is changed, the basic variables change at the rate given by the first column within the basis inverse; i.e., the first variable changes at rate  $(B^{-1})_{11}$ , the second at  $(B^{-1})_{21}$  and so on.

Other results may be derived regarding changes in nonbasic variables. Partially differentiating the objective function equation with respect to a nonbasic variable yields

$$\frac{\partial Z}{\partial x_j} = -(C_B B^{-1} a_j - c_j) \quad j, \text{NB}$$

This shows that the expected marginal cost of increasing a nonbasic variable equals the negative of  $C_B B^{-1} a_j - c_j$ , a consequence the  $C_B B^{-1} a_j - c_j$  term is usually called reduced cost. The marginal effect of changes in the nonbasic variables on the basic variables is obtained by differentiating. This yields

$$\frac{\partial X_B}{\partial x_j} = -B^{-1} a_j \quad j, \text{NB}$$

which shows that the marginal effect of the nonbasic variables on the basic variable is minus  $B^{-1} a_j$ . The  $B^{-1}$  constitutes a translation from the original resource use space (i.e.,  $a_j$ ) into the basic variables space and tells us how many units of each basic variable are removed with a marginal change in the nonbasic variable. We can also use these results to further interpret the  $\partial Z / \partial x_j$  equation. The marginal revenue due to increasing a nonbasic variable is equal to its direct revenue ( $c_j$  the objective function coefficient) less the value of the basic variables ( $C_B$ ) times the amount of the basic variables diverted ( $B^{-1} a_j$ ). Thus, this equation takes into account

both the direct effect from increasing  $x_j$  plus the substitution effect for the basic variables.

### 3.3.2 Examples of Solution Interpretation

This set of general interpretations may be applied to the Joe's Van example above. The appropriate mathematical expressions for each of the four items are as follows.

$$\begin{array}{l}
 \frac{\partial Z}{\partial b} = C_B B^{-1} [1700 \ 2000] \begin{bmatrix} 5 & 1/5 \\ 4 & 1/5 \end{bmatrix} = [500 \ 60] \\
 \frac{\partial X_B}{\partial b} = B^{-1} \begin{bmatrix} 5 & 1/5 \\ 4 & 1/5 \end{bmatrix} \\
 \frac{\partial Z}{\partial X_{NB}} = (C_B B^{-1} A_{NB} - C_{NB}) [1700 \ 2000] \begin{bmatrix} 5 & 1/5 \\ 4 & 1/5 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - [0 \ 0] = [500 \ 61] \\
 \frac{\partial X_B}{\partial X_{NB}} = B^{-1} a_{NB} \begin{bmatrix} 5 & 1/5 \\ 4 & 1/5 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 5 & 1/5 \\ 4 & 1/5 \end{bmatrix}
 \end{array}$$

Note the first expression, which gives the partial of  $Z$  with respect to  $b$ , tells how the objective function changes when the right hand sides change. Thus, if the capacity limit was changed upward from 12, one would expect the objective function to increase \$500 per unit. Similarly if the second right hand side or the labor limit was increased upwards from 280 then one would expect a return of \$60 per hour. The second expression indicates the anticipated change in the values of the basic variables when the right hand sides are changed; the basic variables in the model are arranged with  $X_{fine}$  being first and  $X_{fancy}$  being second. The first column of the basis inverse corresponds to what happens if the van capacity right hand side is changed; whereas, the second column corresponds to what happens if the labor right hand side is changed. Thus, if capacity was expanded to 13, one would expect to produce 5 more fine vans and 4 less fancy vans. Similarly, if labor was expanded, the number of fine vans would decrease by 1/5 per unit and the number of fancy vans would increase by 1/5. The particular signs of these tradeoffs are caused by the original data. Fancy vans use more labor than fine vans. Thus, when capacity is expanded, more fine vans are made since they use labor more intensively while, if labor is increased, one makes more fancy vans.

Now let us examine the effects of changes on the objective function when the nonbasic variables are altered. In this problem we have two nonbasic variables which are the slack variables for the two resources. The effect of increasing the nonbasics is a \$500 decrease if we increase slack capacity, and a \$60 decrease if we increase slack labor. This is exactly the opposite of the resource values discussed above, since the consequence of increasing the slacks is the same as that of decreasing the resource endowments.

The interpretation of the basis inverse also allows us to get further information about the interpretation of the change in the objective function when the right hand sides have changed. Namely, if changing capacity causes five more fine vans to be produced (each worth \$1700, leading to a \$8500 increase but, four less fancy vans worth \$8000) the net effect then is a \$500 increase which equals the shadow price. Similarly, the labor change causes \$400 more worth of fancy vans to be produced but \$340 less fine vans for a net value of \$60. Overall, this shows an important property of linear programming. The optimal solution information contains information about how items substitute. This substitution information is driven by the relative uses of the constraint resources by each of the alternative activities. This is true in more complex linear programming solutions.

### **3.3.3 Finding Limits of Interpretation**

The above interpretations only hold when the basis remains feasible and optimal. Ranging analysis is the most widely utilized tool for analyzing how much a linear program can be altered without changing the interpretation of the solution. Ranging analysis deals with the question: what is the range of values for a particular parameter for which the current solution remains optimal? Ranging analysis of right-hand-side ( $b_i$ ) and objective function coefficients ( $c_j$ ) is common; many computer programs available to solve LP problems have options to conduct ranging analyses although GAMS does not easily support such features (See chapter 19 for details).

#### 3.3.3.1 Ranging Right-Hand-Sides

Let us study what happens if we alter the right hand side (RHS). To do this let us write the new RHS in terms of the old RHS, the size of the change and a direction of change,

$$b_{\text{new}} = b_{\text{old}} + 2r$$

where  $b_{\text{new}}$  is the new RHS,  $b_{\text{old}}$  is the old RHS, 2 is a scalar giving the magnitude of the change and  $r$  is the direction of change. Given an  $r$  vector, the resultant values for the basic variables and objective function are

$$\begin{aligned} X_B &= B^{-1}b_{\text{new}} = B^{-1}(b_{\text{old}} + 2r) = B^{-1}b_{\text{old}} + 2B^{-1}r \\ Z &= C_B B^{-1}b_{\text{new}} = C_B B^{-1}(b_{\text{old}} + 2r) = C_B B^{-1}b_{\text{old}} + 2C_B B^{-1}r \end{aligned}$$

while  $C_B B^{-1}a_j - c_j$  is unchanged. The net effect is that the new solution levels are equal to the old solution levels plus  $2B^{-1}r$  and the new objective function is the old one plus  $2C_B B^{-1}r$ . For the basis to still hold the basic variables must remain nonnegative as must the reduced costs ( $C_B B^{-1}a_j - c_j$ ). However, since the reduced costs are unaltered we must only worry about the basic variables. Thus the condition for  $X_B$  can be written with non-negativity imposed

$$X_B = B^{-1}b_{\text{new}} = B^{-1}b_{\text{old}} + 2B^{-1}r \geq 0$$

and merits further examination in terms of the admissible value of 2.

The above expression gives a simultaneous set of conditions for each basic variable for which one can solve those conditions. In that case, one gets two cases which arise across the set of conditions depending on the sign of individual elements in  $B^{-1}r$ .

$$2 \leq \frac{(B^{-1}b_{\text{old}})_i}{(B^{-1}r)_i}, \text{ where } (B^{-1}r)_i > 0$$

and

$$2 \# \quad \& \frac{(B^{-1}b_{old})_i}{(B^{-1}r)_i}, \text{ where } (B^{-1}r)_i < 0$$

much as in the row minimum rule where positive values of  $B^{-1}r$  limit how negative  $z$  can be and negative numbers limit how positive  $z$  can become. This result shows the range over which  $z$  can be altered with the basis remaining optimal.

Example

Suppose in the Joe's van factory example we wished to change the first right hand side.

Ordinarily, if one wishes to change the  $i^{\text{th}}$  RHS, then  $r$  will be a vector with all zeros except for a one in the  $i^{\text{th}}$  position, as illustrated below

$$r = \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \quad \text{with } i^{\text{th}} \text{ element}$$

Thus

$$b_{\text{new}} = \begin{bmatrix} 12 \\ 280 \end{bmatrix} + 2 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 12 + 2 \\ 280 \end{bmatrix}$$

The resultant values of  $X_B$  becomes

$$X_{B_{\text{new}}} = \begin{bmatrix} 4 \\ 8 \end{bmatrix} + 2 \begin{bmatrix} 5 & 1/5 \\ 4 & 1/5 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 4 \\ 8 \end{bmatrix} + 2 \begin{bmatrix} 5 \\ 4 \end{bmatrix} = \begin{bmatrix} 14 \\ 16 \end{bmatrix}$$

which implies

$$0.8 \leq \frac{4}{5} + 2 \leq \frac{8}{4} + 2$$

Therefore the first right hand side can be changed up by 2 or down by 0.8 without the basis changing. Note that during this alteration the solution  $(B^{-1}b)$  does change, but  $B^{-1}$  does not. Furthermore, this gives a range of values for  $b_1$  for which the marginal value of the resource  $(C_B B^{-1})$  remains the same.

This approach also encompasses a generalization of the RHS ranging problem. Namely, suppose we wish to alter several RHS's at the same time. In this case, the change vector  $(r)$  does not have one entry but rather several. For example, suppose in Joe's van that Joe will add both capacity and an employee. In that case the change vector would look like the following:

$$b_{\text{new}} = \begin{bmatrix} 12 \\ 280 \end{bmatrix} + 2 \begin{bmatrix} 1 \\ 40 \end{bmatrix} = \begin{bmatrix} 14 \\ 360 \end{bmatrix}$$

### 3.3.3.2 Ranging Objective Function Coefficients

The analysis of ranging objective function coefficients is conceptually similar to RHS ranging. We seek to answer the question: what is the range of values for a particular objective function coefficient for which the current basis is optimal?

To examine this question, write the new objective function as the old objective function plus  $(\Delta)$ , which is a change magnitude, times  $T$  which is a direction of change vector.

$$C_{\text{new}} = C_{\text{old}} + (\Delta) T$$

Simultaneously, one has to write an expression for the objective function coefficients of the basic variables

$$C_{B_{\text{new}}} = C_{B_{\text{old}}} + (T_B$$

where  $T_B$  gives the way the  $C_B$ 's are altered. Subsequently, one can reexpress the restriction that the reduced cost values must be nonnegative as

$$(C_B B^{-1} a_j + c_j)_{\text{new}} = C_{B_{\text{new}}} B^{-1} a_j + c_{j_{\text{new}}} \geq 0$$

which reduces to

$$(C_B B^{-1} a_j + c_j)_{\text{new}} = (C_B B^{-1} a_j + c_j)_{\text{old}} + ((T_B B^{-1} a_j + T_j) \geq 0$$

In turn, we discover for nonbasic variables

$$(C_B B^{-1} a_j + c_j$$

while for basic variables

$$\leq \frac{(C_B B^{-1} a_j + c_j)_{\text{old}}}{(T_B B^{-1} a_j + T_j)}, \text{ where } (T_B B^{-1} a_j + T_j) < 0.$$

$$\geq \frac{(C_B B^{-1} a_j + c_j)}{(T_B B^{-1} a_j + T_j)}, \text{ where } (T_B B^{-1} a_j + T_j) > 0$$

### Example

Suppose in our example problem we want to alter the objective function on  $X_{\text{fancy}}$  so it equals

2,000 +  $\lambda$ . The setup then is

$$C_{\text{new}} = [2000 \ 1700 \ 0 \ 0] + (\lambda [1 \ 0 \ 0 \ 0])$$

and

$$T_B = [0 \ 1]$$

so  $C_{B_{\text{new}}} B^{-1} A - C_{\text{new}}$  for the nonbasics equals



$$500 - 60\% \left( \begin{bmatrix} 0 & 1 \\ 4 & 1/5 \end{bmatrix} \begin{bmatrix} 5 & 1/5 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) + 500 - 60\% (4 \quad 1/5) \quad 0$$

which implies  $-300 \leq \Delta \leq 125$  or that the basis is optimal for any objective function value for  $X_{\text{fancy}}$  between 2125 and 1700. This shows a range of prices for  $X_{\text{fancy}}$  for which its optimal level is constant.

### 3.3.3.3 Changes in the Technical Coefficient Matrix

The above analysis examined changes in the objective function coefficients and right hand sides. It is possible that the technical coefficients of several decision variables may be simultaneously varied. This can be done simply if all the variables are nonbasic. Here we examine incremental changes in the constraint matrix. For example, a farmer might purchase a new piece of equipment which alters the labor requirements over several crop enterprises which use that equipment. In this section, procedures which allow analysis of simultaneous incremental changes in the constraint matrix are presented.

Consider a linear programming problem

$$\begin{aligned} \text{Max } Z &= CX \\ \text{s.t. } \bar{A}X &= b \\ X &\geq 0 \end{aligned}$$

where the matrix of the technical coefficients is given by

$$\bar{A} = A + M$$

where  $\bar{A}$ ,  $A$ , and  $M$  are assumed to be  $m \times n$  matrices. Suppose the matrix  $M$  indicates a set simultaneous changes to be made in  $A$  and that the problem solution is nondegenerate, possessing an unique optimal solution. Then the expected change in the optimal value of the objective function given  $M$  is

$$Z_{\text{new}} - Z_{\text{old}} = -U^*MX^*$$

where  $X^*$  and  $U^*$  are the optimal decision variable values ( $B^{-1}b$ ) and shadow prices ( $C_B B^{-1}$ ) for the unaltered original problem.

The equation is an approximation which is exact when the basis does not change. See Freund(1985) for its derivation and further discussion. Intuitively the equation can be explained as follows: since  $M$  gives the per unit change in the resource use by the variables, then  $MX^*$  gives the change in the resources used and  $U^*MX^*$  then gives an approximation of the value of this change. Further, if  $M$  is positive, then more resources are used and the  $Z$  value should go down so a minus is used. McCarl, et al., (1990) have investigated the predictive power of this equation and conclude it is a good approximation for the case they examined.

### Illustrative Example

To illustrate the procedure outlined in the preceding section, consider the Joe's van shop model and suppose we wish to consider the effect of an equal change in the labor coefficients. For a change equal 2, the problem becomes

$$\begin{array}{rcllcl}
 \text{Max } Z(2) & = & 2000X_{\text{fancy}} & + & 1700X_{\text{fine}} & & & \\
 \text{s.t.} & & X_{\text{fancy}} & + & X_{\text{fine}} & + & S_1 & = 12 \\
 & & (25 + 2)X_{\text{fancy}} & + & (20 + 2)X_{\text{fine}} & + & S_2 & = 280 \\
 & & X_{\text{fancy}} & & X_{\text{fine}} & & S_1, S_2 & \geq 0
 \end{array}$$

For no change ( $2 = 0$ ), the optimal solution to this problem is

$$\begin{aligned}
 X & = B^{-1}b = \begin{bmatrix} 4 \\ 8 \end{bmatrix} \\
 U & = C_B B^{-1} = [500 \quad 60]
 \end{aligned}$$

with the optimal value of the objective function equal to 22,800, our change matrix in this case is

$$M = \begin{bmatrix} 0 & 0 \\ 2 & 2 \end{bmatrix}$$

Thus, the change in the value of the objective function is given by

$$Z_{\text{new}} - Z_{\text{old}} = U^* \Delta B^{-1} b$$

$$= \begin{bmatrix} 0 & 0 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} 4 \\ 8 \end{bmatrix} = 720$$

Suppose the labor requirement is reduced by 1 hour for both vans so that  $b = -1$ , then the anticipated increase in the objective function that would result from using the new machine is

$$\Delta Z = -U^* \Delta b = 720$$

a resolution of the revised problem shows the objective function changes by 720.

### 3.3.4 Finding the Solution

As shown above, the linear programming solution contains a lot of information relative to the ways the objective function and basic variables change given changes in parameters. However, not all this information is included in an optimal solution as reported by modeling systems such as GAMS. Consider the following problem

$$\begin{array}{llllll} \text{Max } Z & = & 3x_1 & + & 2x_2 & + & 0.5x_3 & & & \text{(ZROW)} \\ & & x_1 & + & x_2 & + & x_3 & = & 10 & \text{(CONSTRAIN 1)} \\ & & x_1 & + & x_2 & & & = & 3 & \text{(CONSTRAIN 2)} \\ & & x_1, & & x_2 & & & \geq & 0. & \end{array}$$

The GAMS solution information for this problem appears in Table 3.1. The optimal objective function value equals 26.5. Then GAMS gives information on the equations. For this problem, there are 3 equations as named in the parenthetical statements above. For each equation information is given on the lower limit (labeled LOWER), value of  $AX^*$  (labeled LEVEL), upper limit (labeled UPPER), and shadow price  $C_B B^{-1}$  (labeled MARGINAL). The objective function row (ZROW) does not contain interesting information. The constraint equations show there is a) no lower bound on the first equation (there would be if it were  $\geq$ ) b) the

left hand side equals 10 ( $AX^*$ ) and c) the right hand side is 10 (UPPER) while the shadow price is 2.5 (MARGINAL). Similar information is present for the second equation.

Turning to the variables, the solution table gives the variable name, lower bound (LOWER), optimal level (LEVEL), upperbound (UPPER) and reduced cost (MARGINAL). The solution shows  $x_1 = 6.5$  and  $x_2 = 3.5$  while the cost of forcing  $x_3$  into the solution is estimated to be \$2.00 per unit. We also see the objective function variable (Z) equals 26.5. The solution information also indicates if an unbounded or infeasible solution arises.

GAMS output does not provide access to the  $B^{-1}$  or  $B^{-1}a_j$  matrices. This is a mixed blessing. A 1000 row model would have quite large  $B^{-1}$  and  $B^{-1}a_j$  matrices, but there are cases where it would be nice to have some of the information. None of the GAMS solvers provide access to this data.

### 3.3.5 Alternative Optimal and Degenerate Cases

Linear programming does not always yield a unique primal solution or resource valuation. Non-unique solutions occur when the model solution exhibits degeneracy and/or an alternative optimal.

Degeneracy occurs when one or more of the basic variables are equal to zero. Degeneracy is a consequence of constraints which are redundant in terms of their coefficients for the basic variables. Mathematically, given a problem with  $M$  rows and  $N$  original variables, and  $M$  slacks, degeneracy occurs when there are more than  $N$  original variables plus slacks that equal zero with less than  $M$  of the original variables and slacks being non-zero.

Most of the discussion in LP texts regarding degeneracy involves whether or not degeneracy makes the problem harder to solve and most texts conclude it does not. Degeneracy also has important implications for resource valuation. Consider for example the following problem:

$$\begin{array}{rcl}
 \text{max} & 100X_1 & + 70X_2 \\
 & X_1 & \# 50 \\
 & & X_2 \# 50 \\
 & X_1 & \% X_2 \# 100
 \end{array}$$

The solution to this problem is degenerate because the third constraint is redundant to the first two. Upon application of the simplex algorithm, one finds in the second iteration that the variable  $X_2$  can be entered in place of the slack variable from either the second or third rows. If  $X_2$  is brought into the basis in place of the second slack, the shadow prices determined are  $(u_1, u_2, u_3) = (100, 75, 0)$ . If  $X_2$  is brought into the basis in place of the third slack, the value of the shadow prices are  $(u_1, u_2, u_3) = (25, 0, 75)$ . These differ depending on whether the second or third slack variable is in the basis at a value of zero. Thus, the solution is degenerate, since a variable in the basis (one of the slacks) is equal to zero (given three constraints there would be three non-zero variables in a non-degenerate solution). The alternative sets of resource values may cause difficulty in the solution interpretation process. For example, under the first case, one would interpret the value of the resource in the second constraint as \$75, whereas in the second case it would interpret nominally as \$0. Here the shadow prices have a direction and magnitude as elaborated in McCarl (1977) (this has been shown numerous times, see Drynan or Gal, Kruse, and Zornig.). Note that decreasing the RHS of the first constraint from 100 to 99 would result in a change in the objective function of 100 as predicted by the first shadow price set, whereas increasing it from 100 to 101 would result in a \$25 increase as predicted by the first shadow price set. Thus, both sets of shadow prices are valid. The degenerate solutions imply multiple sets of resource valuation information with any one set potentially misleading. Both McCarl (1977) and Paris discuss approaches which may be undertaken in

such a case. The underlying problem is that some of the right hand side ranges are zero, thus the basis will

change for any right hand side alterations in one direction.

Another possibility in the simplex algorithm is the case of alternate optimal. An alternative optimal occurs when at least one of the nonbasic variables has a zero reduced cost; i.e.,  $C_B B^{-1} a_j - c_j$  for some  $j \in NB$  equal to zero. Thus, one could pivot, or bring that particular variable in the solution replacing one of the basic variables without changing the optimal objective function value. Alternative optimals also imply that the reduced cost of more than  $M$  variables in a problem with  $M$  constraints are equal to zero. Consider the following problem:

$$\begin{aligned} \text{Max } & 25X_1 + 50X_2 \\ & X_1 + 2X_2 \leq 100 \end{aligned}$$

In this problem the optimal solution may consist of either  $X_1 = 100$  or  $X_2 = 50$  with equal objective function values one or the other of these variables will have zero reduced cost at optimality. Alternative optimals may cause difficulty to the applied modeler as there is more than one answer which is optimal for the problem.

Paris (1981, 1991); McCarl et al. (1977); McCarl and Nelson, and Burton et al., discuss this issue further.

### 3.3.6 Finding Shadow Prices for Bounds on Variables

Linear programming codes impose upper and lower bounds on individual variables in a special way. Many modelers do not understand where upper or lower bound related shadow prices appear. An example of a problem with upper and lower bounds is given below.

$$\begin{aligned} \text{max } & 3X_1 + X_2 \\ \text{s.t. } & X_1 + X_2 \leq 15 \\ & X_1 \leq 10 \\ & X_2 \leq 1 \end{aligned}$$

The second constraint imposes an upper bound on  $X_1$ , i.e.,  $X_1 \leq 10$ , while the third constraint,  $X_2 \leq 1$ , is a

lower bound on  $X_2$ . Most LP algorithms allow one to specify these particular restrictions as either constraints or bounds. Solutions from LP codes under both are shown in Table 3.2.

In the first solution the model has three constraints, but in the second solution the model has only one constraint with the individual constraints on  $X_1$  and  $X_2$  imposed as bounds. Note that in the first solution there are shadow prices associated with constraints two and three. However, this information does not appear in the equation section of the second solution table. A closer examination indicates that while  $X_1$  and  $X_2$  are non-zero in the optimal solution, they also have reduced costs. Variables having both a non-zero value and a non-zero reduced cost are seemingly not in accordance with the basic/nonbasic variable distinction. However, the bounds have been treated implicitly. Variables are transformed so that inside the algorithm they are replaced by differences from their bounds and thus a nonbasic zero value can indicate the variable equals its bound. Thus, in general, the shadow prices on the bounds are contained within the reduced cost section of the column solution. In the example above the reduced costs show the shadow price on the lower bound of  $X_2$  is 1 and the shadow price on the upper bound of  $X_1$  is -3. Notice these are equal to the negative of the shadow prices from the solution when the bounds are treated as constraints.

One basic advantage of considering the upper and lower limits on variables as bounds rather than constraints is the smaller number of rows which are required.

### **3.4 Further Details on the Simplex Method**

The simplex method as presented above is rather idealistic avoiding a number of difficulties. Here we present additional details regarding the basis in use, finding an initial nonnegative basis and some comments on the real LP solution method.

### 3.4.1 Updating the Basis Inverse

In step 5 of the matrix simplex method the basis inverse needs to be changed to reflect the replacement of one column in the basis with another. This can be done interactively using the so-called product form of the inverse (Hadley(1962)). In using this procedure the revised basis inverse ( $B^{-1}$ ) is the old basis inverse ( $B^{-1}$ ) times an elementary pivot matrix ( $P$ ), i.e.,

$$B_{\text{new}}^{-1} = PB_{\text{old}}^{-1}$$

This pivot matrix is formed by replacing the  $I^{*th}$  (where one is pivoting in row  $I^*$ ) column of an identity matrix with elements derived from the column associated with the entering variable. Namely, suppose the entering variable column updated by the current basis inverse has elements

$$a_0^{(i)} = B^{-1}a_0.$$

then the elements of the elementary pivot matrix are

$$P_i(i, i) = 1/a_{i0}^{(i)}$$

$$P_{ki}(i, k) = a_{k0}^{(i)}/a_{i0}^{(i)}, \quad k \neq i$$

Suppose we update the inverse in the Joe's van example problem using product form of the inverse. In the first pivot, after  $X_{\text{fancy}}$  has been identified to enter the problem in row 2, then we replace the second column in an identity matrix with a column with one over the pivot element (the element in the second row of  $B^{-1}a_0$  divided by the pivot element elsewhere. Since  $B^{-1}a_0$  equals  $\begin{bmatrix} 1 \\ 25 \end{bmatrix}$ , the pivot matrix  $P_1$  is

$$P_1 = \begin{bmatrix} 1 & 1/25 \\ 0 & 1/25 \end{bmatrix}$$



and the new  $B^{-1}$  is

$$B_1^{-1} = P_1 B_0^{-1} = \left[ \begin{array}{cc|cc} 1 & 1/25 & 1 & 0 \\ 0 & 1/25 & 0 & 1 \end{array} \right], \quad \left[ \begin{array}{cc} 1 & 1/25 \\ 0 & 1/25 \end{array} \right]$$

$$P_2 = \left[ \begin{array}{cc|cc} 1/(1/5) & 0 & 5 & 0 \\ (4/5)/(1/5) & 1 & 4 & 1 \end{array} \right]$$

Similarly in the second pivot we find the minimum in the first row and have  $B^{-1}a_0 = \begin{bmatrix} 1/5 \\ 4/5 \end{bmatrix}$ , so that in forming  $P_2$ , the first column of an identity matrix was replaced since  $x_5$  will enter as the first element of the basis vector. Multiplication of  $B^{-1}$  by  $P_2$  gives

$$B_2^{-1} = P_2 B_1^{-1} = \left[ \begin{array}{cc} 5 & 0 \\ 4 & 1 \end{array} \right] \left[ \begin{array}{cc|cc} 1 & 1/25 \\ 0 & 1/25 \end{array} \right], \quad \left[ \begin{array}{cc} 5 & 1/5 \\ 4 & 1/5 \end{array} \right]$$

which equals the basis inverse computed above.

### 3.4.2 Advanced Bases

The process of solving a LP is a hunt for the optimal basis matrix. Experience with LP reveals that the simplex method usually requires two or three times as many iterations as the number of constraints to find an optimal basis. This implies that when solving a series of related problems (i.e., changing a price of an input), it may be worthwhile to try to save the basis from one problem and begin the next problem from that particular basis. This is commonly supported in LP solution algorithms and is quite important in applied LP involving sizable models. In a recent study, it took more than thirty-five hours of computer time to obtain an initial basis, but from an advanced basis, a series of related problems with a few changes in parameters could be solved in two hours. Dillon (1970) discusses ways of suggesting a basis for problems that have not previously been solved.

Modeling systems like GAMS do not readily take an advanced basis although one can be attempted by a choice of initial levels for variables (GAMBAS (McCarl (1996)) permits this). However, once an initial model solution has been found, then any additional solutions are computed from an initial basis. Furthermore, an advanced basis can be employed by restarting from a stored file.

### 3.4.3 Finding an Initial Feasible Basis

When an LP problem includes only less-than constraints with non-negative right hand sides, it is straightforward to obtain an initial feasible basis with all non-negative variable values. In that case the slacks form the initial basis and all decision variables are nonbasic, equaling zero, with each slack variable set equal to the RHS ( $s_i = b_i$ ). The initial basis matrix is an identity matrix. In turn, the simplex algorithm is initiated.

However, if one or more: a) negative right hand sides, b) equality constraints, and/or c) greater than or equal to constraints are included, it is typically more difficult to identify an initial feasible basis. Two procedures have evolved to deal with this situation: the Big M method and the Phase I/Phase II method. Conceptually, these two procedures are similar, both imply an inclusion of new, artificial variables, which artificially enlarge the feasible region so an initial feasible basis is present. The mechanics of artificial variables, of the Big M method and the Phase I/ Phase II problem are presented in this section.

Models which contain negative right hand sides, equality and or greater than constraints do not yield an initial feasible solution when all X's are set to zero. Suppose we have the following

$$\begin{array}{ll}
 \text{Max} & CX \\
 \text{s.t.} & RX \# \quad b \\
 & DX \# \quad \& e \\
 & FX ' \quad g \\
 & HX \$ \quad p \\
 & X \$ \quad 0
 \end{array}$$

general problem where b, e, and p are positive.

Conversion of this problem to the equality form requires the addition of slack, surplus and artificial variables. The slack variables ( $S_1$  and  $S_2$ ) are added to the first and second rows (note that while we cover this topic here, most solvers do this automatically). Surplus variables are needed in the last constraint type and give the amount that left hand side (HX) exceeds the right hand side limit (p). Thus, the surplus variables (W) equal  $p - HX$  and the constraint becomes  $HX - W = p$ .

The resultant equality form becomes

$$\begin{array}{rcccccccc}
 \text{Max} & CX & \% & O_1 S_1 & \% & O_2 S_2 & \% & O_4 W \\
 \text{s.t.} & RX & \% & I_1 S_1 & & & & & b \\
 & DX & & & \% & I_2 S_2 & & & \& e \\
 & FX & & & & & & & g \\
 & HX & & & & & \& I_4 W & p \\
 & X, & & S_1, & & S_2, & & W & \$ 0
 \end{array}$$

Where the I's are appropriately sized identity matrices and the O's are appropriately sized vectors of zeros.

Note that when  $X = 0$ , the slacks and surplus variables do not constitute an initial feasible basis. Namely, if  $S_2$  and  $W$  are put in the basis; then assuming  $e$  and  $p$  are positive, the initial solution for these variables are negative violating the non-negativity constraint (i.e.,  $S_2 = -e$  and  $W = -p$ ). Furthermore, there is no apparent initial basis to specify for the third set of constraints ( $FX = g$ ). This situation requires the use of artificial variables. These are variables entered into the problem which permit an initial feasible basis, but need to be removed from the solution before the solution is finalized.

Artificial variables are entered into each constraint which is not satisfied when  $X=0$  and does not have an easily identified basic variable. In this example, three sets of artificial variables are required.

$$\begin{array}{rcccccccc}
\text{Max} & CX & \% & O_1S_1 & \% & O_2S_2 & \% & O_4W \\
& RX & \% & I_1S_1 & & & & & ' & b \\
& DX & & & \% & I_2S_2 & & & \& & I_2A_2 & & ' & \& e \\
& FX & & & & & & & \% & I_3A_3 & & & ' & g \\
& HX & & & & & \& & I_4W & & & \% & I_4A_4 & & ' & p \\
& X, & & S_1, & & S_2, & & W, & & A_2, & & A_3, & & A_4 & \$ & 0
\end{array}$$

Here  $A_2$ ,  $A_3$ , and  $A_4$  are the artificial variables which permit an initial feasible nonnegative basis but which must be removed before a "true feasible solution" is present. Note that  $S_1$ ,  $A_2$ ,  $A_3$ , and  $A_4$  can be put into the initial basis. However, if elements of  $A_3$  are nonzero in the final solution, then the original  $FX = g$  constraints are not satisfied. Similar observations are appropriate for  $A_2$  and  $A_4$ . Consequently, the formulation is not yet complete. The objective function must be manipulated to cause the artificial variables to be removed from the solution. The two alternative approaches reported in the literature are the BIG M method and the Phase I/Phase II method.

### 3.4.3.1 BIG M Method

The BIG M method involves adding large penalty costs to the objective function for each artificial variable. Namely, the objective function of the above problem is written as

$$\text{Max } CX \% O_1S_1 \% O_2S_2 \% O_4W \& M_2A_2 \& M_3A_3 \& M_4A_4$$

where  $M_2$ ,  $M_3$ , and  $M_4$  are conformable sized vectors of large numbers that will cause the model to drive  $A_2$ ,  $A_3$ , and  $A_4$  out of the optimal solution.

An example of this procedure involves the problem

$$\begin{array}{rcllcl}
 \text{Max} & 3x_1 & + & 2x_2 & & \\
 & x_1 & + & 2x_2 & = & 10 \\
 & 2x_1 & + & x_2 & = & 2 \\
 & x_1 & + & x_2 & = & 3 \\
 & x_1 & + & x_2 & = & 1 \\
 & x_1, & & x_2 & \geq & 0
 \end{array}$$

and the model as prepared for the Big M method is in Table 3.3. The optimal solution to this problem is in Table 3.4.

This solution is feasible since  $A_2$ ,  $A_3$ , and  $A_4$  have been removed from the solution. On the other hand, if the right hand side on the second constraint is changed to  $-4$ , then  $A_2$  cannot be forced from the solution and the problem is infeasible. This, with the BIG M method one should note that the artificial variables must be driven from the solution for the problem to be feasible so  $M$  must be set large enough to insure this happens (if possible).

### 3.4.3.2 Phase I/Phase II Method

The Phase I/Phase II method is implemented in almost all computer codes. The procedure involves the solution of two problems. First, (Phase I) the problem is solved with the objective function replaced with an alternative objective function which minimizes the sum of the artificial variables, i.e.,

$$\text{Min } L_2A_2 + L_3A_3 + L_4A_4$$

where  $L_i$  are conformably sized row vectors of ones.

If the Phase I problem has a nonzero objective function (i.e., not all of the artificials are zero when their sum has been minimized), then the problem does not have a feasible solution. Note this means the reduced cost information in an infeasible problem can correspond to this modified objective function.

Otherwise, drop the artificial variables from the problem and return to solve the real problem (Phase II) using the Phase I optimal basis as a starting basis and solve using the normal simplex procedure.

The addition of the slack, surplus and artificial variables is performed automatically in almost all solvers including all that are associated with GAMS.

### **3.4.4 The Real LP Solution Method**

The above material does not fully describe how a LP solution algorithm works. However, the algorithm implemented in modern computer codes, while conceptually similar to that above is operationally quite different. Today some codes are using interior point algorithms combined with the simplex method (for instance, OSL, Singhal et al.). Codes also deal with many other things such as compact data storage, basis reinversion, efficient pricing, and round-off error control (e.g., see Orchard-Hays or Murtagh).

In terms of data storage, algorithms do not store the LP matrix as a complete  $MXN$  matrix. Rather, they exploit the fact that LP problems often be sparse, having a small number of non-zero coefficients relative to the total possible number, by only storing non-zero coefficients along with their column and row addresses. Further, some codes exploit packing of multiple addresses into a single word and economize on the storage of repeated numerical values (for in-depth discussion of data storage topics see Orchard-Hays or Murtagh).

Perhaps the most complex part of most modern day LP solvers involves inversion. As indicated above, the  $B^{-1}$  associated with the optimal solution is needed, but in forming  $B^{-1}$  the code usually performs more iterations than the number of constraints. Thus, the codes periodically construct the basis inverse from the original data. This is done using product form of the inverse; but this also involves such diverse topics as LU decomposition, reduction of a matrix into lower triangular form and matrix factorization. For discussion in these topics see Murtagh.

LP codes often call the formation of reduced costs the pricing pass and a number of different approaches have been developed for more efficient computation of pricing (see Murtagh for discussion).

Finally, LP codes try to avoid numerical error. In computational LP, one worries about whether numbers are really non-zero or whether rounding error has caused fractions to compound giving false non-zeros. Solver implementations usually make extensive use of tolerances and basis reinversion schemes to control such errors. Murtagh and Orchard-Hays discuss these.

The purpose of the above discussion is not to communicate the intricacies of modern LP solvers, but rather to indicate that they are far more complicated than the standard implementation of the simplex algorithm as presented in the first part of the chapter.

## References

- Bazarrar, M.S. and J.J. Jarvis, and H.D. Sherali. Linear Programming and Network Flows. New York: John Wiley & Sons, 1990.
- Burton, R.O., J.S. Gidley, B.S. Baker and K.J. Reda-Wilson. "Nearly Optimal Linear Programming Solutions: Some Conceptual Issues and a Farm Management Application." American Journal of Agricultural Economics. 69(1987):813-818.
- Dantzig, G.B. Linear Programming and Extension. Princeton, New Jersey: Princeton University Press, 1963.
- Dillon, M. "Heuristic Selection of Advanced Bases for a Class of Linear Programming Models." Operation Research. 18(1970):90-100.
- Drynan, R.G. "Forum on Resolving Multiple Optima in Linear Programming." Review of Marketing and Agricultural Economics. 54(1986):31-35.
- Freund, R.M. "Postoptimal Analysis of a Linear Program Under Simultaneous Changes in Matrix Coefficients." Mathematical Programming Studies. 24(1985):1-13.
- Gal, T., H.J. Kruse, and P. Zornig. "Survey of Solved and Open Problems in the Degeneracy Phenomenon." Mathematical Programming. 42(1988):125-133.
- Hadley, G. Linear Programming. Reading, Mass.: Addison-Wesley Publishing Co., 1962.
- Karmarkar, N. "A New Polynomial-Time Algorithm for Linear Programming." Combinatorica. 4(1984):373-395.
- McCarl, B.A. "Degeneracy, Duality and Shadow Prices in Linear Programming." Canadian Journal of Agricultural Economics. 25(1977):70-73.
- McCarl, B.A., W.V. Candler, D.H. Doster, and P. Robbins. "Experiences with Farmer Oriented Linear Programming for Crop Planning." Canadian Journal of Agricultural Economics. 25(1977):17-30.
- McCarl, B.A., D.E. Kline, and D.A. Bender. "Improving on Shadow Price Information for Identifying Critical Farm Machinery." American Journal of Agricultural Economics. 72(1990):582-588.
- McCarl, B.A. and C.H. Nelson. "Multiple Optimal Solutions in Linear Programming Models: Comment." American Journal of Agricultural Economics. 65(1983):181-83.
- Murtagh, B.A. Advanced Linear Programming. McGraw-Hill Book Co., New York, 1981.
- Orchard-Hays, W. Advanced Linear- Programming Computing Techniques. McGraw-Hill, Inc: New York, 1968.
- Optimization Subroutine Library (OSL). IBM Guide and Reference, Release 2. Ed. J.A. George and J.W.H.



Liu. IBM Corporation, 1990-91.

Paris, Q. "Multiple Optimal Solutions in Linear Programming Models." American Journal of Agricultural Economics. 63(1981):724-27.

Paris, Q. An Economic Interpretation of Linear Programming. Iowa State University Press: Ames, Iowa, 1991.

Singhal, J.A., R.E. Marsten, and T.L. Morin. "Fixed Order Branch-and-Bound Methods for Mixed-Integer Programming: The ZOOM System." Department of Management Information Systems, University of Arizona, Tucson, AZ, 85721, December, 1987.

**Table 3.1. GAMS Solution of Example Model**

SOLVE SUMMARY

```

MODEL      PROBLEM      OBJECTIVE Z
TYPE       LP           DIRECTION MAXIMIZE
SOLVER     MINOS5       FROM LINE 37

**** SOLVER STATUS      1 NORMAL COMPLETION
**** MODEL STATUS       1 OPTIMAL
**** OBJECTIVE VALUE    26.5000
  
```

EXIT -- OPTIMAL SOLUTION FOUND

	LOWER	LEVEL	UPPER	MARGINAL
--- EQU ZROW	.	.	.	1.000
--- EQU CONSTRAIN1	-INF	10.000	10.000	2.500
--- EQU CONSTRAIN2	-INF	3.000	3.000	0.500

```

ZROW      OBJECTIVE FUNCTION
CONSTRAIN1 FIRST CONSTRAINT
CONSTRAIN2 SECOND CONSTRAINT
  
```

	LOWER	LEVEL	UPPER	MARGINAL
--- VAR X1	.	6.500	+INF	.
--- VAR X2	.	3.500	+INF	.
--- VAR X3	.	.	+INF	-2.000
--- VAR Z	-INF	26.500	+INF	.

```

X1      FIRST VARIABLE
X2      SECOND VARIABLE
X3      THIRD VARIABLE
Z       OBJECTIVE FUNCTION
  
```

**Table 3.2. Solution to Bounded Example with Bounds Imposed as Constraints and as Bounds**

		Solved with Constraints					
Variable	Value	Reduced Cost	Status	Equation	Level	Shadow Price	Status
$X_1$	10	0	Basic	1	4	0	Basic
$X_2$	1	0	Basic	2	0	3	
				3	0	-1	

		Solved with Bounds					
Variable	Value	Reduced Cost	Status	Equation	Level	Shadow Price	Status
$X_1$	10	-3	(U)	1	4	0	Basic
$X_2$	1	1	(L)				

**Table 3.3. The Model as Ready for the Big M Method**

Max	$3x_1$	+	$2x_2$	+	$0S_1$	+	$0S_2$	+	$0W$	-	$99A_2$	-	$99A_3$	-	$99a_4$	=	10
	$x_1$	+	$x_2$	+	$S_1$												
	$x_1$	-	$x_2$		$S_2$						$A_2$						= -2
	$-x_1$	+	$x_2$									+	$A_3$				= 3
	$x_1$	+	$x_2$						$W$					+	$A_4$		= 1
	$x_1$	,	$x_2$	,	$S_1$	,	$S_2$	,	$W$	,	$A_2$	,	$A_3$	,	$A_4$	§	0

**Table 3.4. Solution to the Big M Problem**

Variable	Value	Reduced Cost	Equation	Shadow Price
$x_1$	1.333	0	1	1.667
$x_2$	4.333	0	2	0
$S_1$	0	1.667	3	-1.333
$S_2$	1	0	4	0
W	4.667			
$A_2$	0	-99		
$A_3$	0	-97.667		
$A_4$	0	-99		

## CHAPTER IV: DUALITY IN LINEAR PROGRAMMING

Economic theory indicates that scarce (limited) resources have value. For example, prime agricultural land is limited and has value (a rental price). On the other hand, air is effectively unlimited and therefore does not have a market value. In LP models, limited resources are allocated, so they should be valued. Whenever we solve an LP problem, we solve two problems: the primal resource allocation problem, and the dual resource valuation problem. This chapter covers the resource valuation, or as it is commonly called, the Dual LP problem and its relationship to the original, primal, problem.

The study of duality is very important in LP. Knowledge of duality allows one to develop increased insight into LP solution interpretation. Also, when solving the dual of any problem, one simultaneously solves the primal. Thus, duality is an alternative way of solving LP problems. However, given today's computer capabilities, this is an infrequently used aspect of duality. Therefore, we concentrate on the study of duality as a means of gaining insight into the LP solution. We will also discuss the ways that primal decision variables place constraints upon the resource valuation information.

### 4.1 Basic Duality

The Primal problem can be written as:

$$\begin{aligned} \text{Max} \quad & \sum_j c_j x_j \\ \text{s.t.} \quad & \sum_j a_{ij} x_j \leq b_i, \text{ for all } i \\ & x_j \geq 0, \text{ for all } j \end{aligned}$$

Associated with this primal problem is a dual resource valuation problem. The dual of the above problem is

$$\begin{aligned} \text{Min} \quad & \sum_i u_i b_i \\ \text{s.t.} \quad & \sum_i u_i a_{ij} \leq c_j \text{ for all } j \\ & u_i \geq 0 \text{ for all } i \end{aligned}$$

where  $u_i$  are the dual variables.

If the primal problem has  $n$  variables and  $m$  resource constraints, the dual problem will have  $m$  variables and  $n$  constraints. There is a one-to-one correspondence between the primal constraints and the dual variables; i.e.,  $u_1$  is associated with the first primal constraint,  $u_2$  with the second primal constraint, etc. As we demonstrate later, dual variables ( $u_i$ ) can be interpreted as the marginal value of each constraint's resources. These dual variables are usually called shadow prices and indicate the imputed value of each resource. A one-to-one correspondence also exists between the primal variables and the dual constraints;  $x_1$  is associated with the first dual constraint ( $\sum_i u_i a_{i1} \leq c_1$ ),  $x_2$  is associated with the second dual constraint ( $\sum_i u_i a_{i2} \leq c_2$ ), etc.

An example aids in explaining the dual. Consider the primal model:

$$\begin{array}{llll}
 \text{Max} & 40x_1 & + & 30x_2 & & \text{(profits)} \\
 \text{s.t.} & x_1 & + & x_2 & \leq & 120 \quad \text{(land)} \\
 & 4x_1 & + & 2x_2 & \leq & 320 \quad \text{(labor)} \\
 & x_1 & , & x_2 & \geq & 0
 \end{array}$$

The associated dual problem is

$$\begin{array}{llll}
 & \text{(land)} & \text{(labor)} & \\
 \text{Min} & 120u_1 & + & 320u_2 & \\
 \text{s.t.} & u_1 & + & 4u_2 & \leq 40 \quad (x_1) \\
 & u_1 & + & 2u_2 & \leq 30 \quad (x_2) \\
 & u_1 & , & u_2 & \geq 0
 \end{array}$$

The dual problem economic interpretation is important. The variable  $u_1$  gives the marginal value of the first resource, or land. Variable  $u_2$  gives the marginal value of the second resource, or labor in this case. The first dual constraint restricts the value of the resources used in producing a unit of  $x_1$  to be greater than or equal to the marginal revenue contribution of  $x_1$ . In the primal problem  $x_1$  uses one unit of land and four units of labor, returning \$40, while the dual problem requires land use times its marginal value ( $1u_1$ ) plus labor use

times its marginal value ( $4u_2$ ) to be greater than or equal to the profit earned when one unit of  $x_1$  is produced (40). Similarly, constraint 2 requires the marginal value of land plus twice the marginal value of labor to be greater than or equal to 30, which is the amount of profit earned by producing  $x_2$ . Thus, the dual variable values are constrained such that the marginal value of the resources used by each primal variable is no less than the marginal profit contribution of that variable.

Now suppose we examine the objective function. This function minimizes the total marginal value of the resources on hand. In the example, this amounts to the land endowment times the marginal value of land ( $120u_1$ ) plus the labor endowment times the marginal value of labor ( $320u_2$ ). Thus, the dual variables arise from a problem minimizing the marginal value of the resource endowment subject to constraints requiring that the marginal value of the resources used in producing each product must be at least as great as the marginal value of the product. This can be viewed as the problem of a resource purchaser in a perfectly competitive market. Under such circumstances, the purchaser would have to pay at least as much for the resources as the value of the goods produced using those resources. However, the purchaser would try to minimize the total cost of the resources acquired.

The resultant dual variable values are measures of the marginal value of the resources. The objective function is the minimum value of the resource endowment. Any slack in the constraints is the amount that cost exceeds revenue.

## **4.2 Primal-Dual Solution Inter-Relationships**

Several relationships exist between primal and dual solutions which are fundamental to understanding duality and interpreting LP solutions.



<u>Primal</u>	<u>Dual</u>
Max $CX$	Min $U^T b$
$AX \leq b$	$U^T A \leq C$
$X \geq 0$	$U \geq 0$

First, let us introduce some notation. The primal dual pair of LP problems in matrix form is  
 Now let us examine how the problems are related.

#### 4.2.1 Objective Function Interrelationships

Suppose we have any two feasible primal and dual solutions  $X^*$ ,  $U^*$  and we want to determine the relationship between the objective functions  $CX^*$  and  $U^{*T} b$ . We know the feasible solutions must satisfy

$$\begin{aligned} AX^* &\leq b & \text{and} & & U^{*T} A &\leq C \\ X^* &\geq 0 & & & U^* &\geq 0 \end{aligned}$$

To determine the relationship, we take the above constraint inequalities (not the non-negativity conditions) and premultiply the left one by  $U^{*T}$  while postmultiplying the right one by  $X^*$ .

$$\begin{aligned} U^{*T} AX^* &\leq C X^* \\ U^{*T} AX^* &\leq U^{*T} b \end{aligned}$$

Noting that the term  $U^{*T} AX^*$  is common to both inequalities we get

$$\begin{aligned} C X^* &\leq U^{*T} AX^* \leq U^{*T} b \\ C X^* &\leq U^{*T} b \end{aligned}$$

This shows that the dual objective function value is always greater than or equal to the primal objective function value for any pair of feasible solutions.

#### 4.2.2 Constructing Dual Solutions

Now suppose we explore the construction of an optimal solution to the dual problem, given a primal optimal solution. Suppose an optimal primal solution is given by  $X_B^* = B^{-1}b$  and  $X_{NB}^* = 0$ . This solution must be feasible; i.e.,  $X_B^* = B^{-1}b \geq 0$ . It also must have  $X_{NB} \geq 0$  and must satisfy nonnegative reduced cost for the nonbasic variables  $C_B B^{-1}A_{NB} - C_{NB} \geq 0$ . Given this, suppose we try  $U^{(1)} = C_B B^{-1}$  as a potential dual solution.

First, let us investigate whether this is feasible. To be feasible, we must have  $U^{(1)} A \leq C$  and  $U^* \geq 0$ .

If we set  $U^{(1)} = C_B B^{-1}$ , then we know  $U^{(1)} A_{NB} - C_{NB} \geq 0$  because at optimality this is exactly equivalent to the reduced cost criteria, i.e.,  $C_B B^{-1}A_{NB} - C_{NB} \geq 0$ . Further, we know for the basic variables the reduced cost is  $C_B B^{-1}A_B - C_B = C_B B^{-1}B - C_B = C_B - C_B = 0$ , so  $U^{(1)} B = C_B$ . By unifying these two statements, we know when  $U^* = C_B B^{-1}$  then  $U^{(1)} A \leq C$  for all primal variables.

Now we need to know if  $U \geq 0$ . The problem includes slacks. Thus,  $A$  includes an identity matrix for the slacks and  $C$  includes 0's for the slack variables. Thus, a part of the  $UA \leq C$  covers the slacks and since we know that  $U^{(1)} A_S \leq C_S$  where  $A_S$  and  $C_S$  are the portions of  $A$  and  $C$  relevant to the slacks. Substituting in the known structure of  $A_S$  and  $C_S$ , i.e.,  $A_S = I$  and  $C_S = 0$  yields  $U^{(1)} \leq 0$  or  $U \geq 0$ . So the  $U$ 's are non-negative. Thus,  $U^{(1)} = C_B B^{-1}$  is a feasible dual solution.

Now the question becomes, is this choice optimal? In this case the primal objective function  $Z$  equals  $CX^* = C_B X_B^* + C_{NB} X_{NB}^*$  and since  $X_{NB}^*$  equals zero, then  $Z = C_B B^{-1}b + C_{NB} 0 = C_B B^{-1}b$ . Simultaneously, the dual objective equals  $U^{(1)} b = C_B B^{-1}b$  which equals the primal objective. Therefore, the primal and dual objectives are equal at optimality. Furthermore, since the primal objective must be less than or equal to the dual objective for any feasible pair of solutions and since they are equal, then clearly  $CX^*$  cannot get any

larger nor can  $U^* b$  get any smaller, so they must both be optimal. Therefore,  $C_B B^{-1} b$  is an optimal dual solution. This demonstration shows that given the solution from the primal the dual solution can simply be computed without need to solve the dual problem.

In addition given the derivation in the last chapter we can establish the interpretation of the dual variables. In particular, since the optimal dual variables equal  $C_B B^{-1} b$  (which are called the primal shadow prices) then the dual variables are interpretable as the marginal value product of the resources since we showed

$$\frac{\Delta Z}{\Delta b} = C_B B^{-1} b = U^* b.$$

### 4.2.3 Complementary Slackness

Yet another, interrelationship between the primal and dual solutions is the so called complementary slackness relation. This states that given optimal values  $U^*$  and  $X^*$  then

$$\begin{aligned} U^* (b - AX^*) &= 0 \\ (U^* A - C) X^* &= 0. \end{aligned}$$

This result coupled with the primal and dual feasibility restrictions ( $U \geq 0$ ;  $UA \leq C$ ;  $AX \leq b$ ;  $X \geq 0$ ) implies that (in the absence of degeneracy) for each constraint, either the resource is fully used ( $b_i - (AX)_i = 0$ ) with the associated dual variable ( $U_i^*$ ) nonzero, or the dual variable is zero with associated unused resources ( $b_i - (AX)_i$ ) being nonzero. Alternatively, for each variable (again ignoring degeneracy) at optimality, either the variable level ( $X_j$ ) is non-zero with zero reduced cost ( $(U^* A)_j - c_j = 0$ ) or the variable is set to zero with a non-zero reduced cost.

This result may be proven using matrix algebra. Given optimal primal ( $X^*$ ) and dual ( $U^*$ ) solutions

$$\text{Let } \pi = U'(b - AX^*) \\ \lambda = (U' A - C)X^*$$

Now suppose we examine  $\pi + \lambda$ . Taking  $\pi + \lambda$ , we get  $U'(b - AX^*) + (U' A - C)X^*$  which equals  $U' b -$

$U' AX^* + U' AX^* - CX^* = U' b - CX^*$ . We know this equals zero at optimality. Further, we know that

both  $\pi$  and  $\lambda$  will be nonnegative, since  $AX^* \leq b$ ,  $U' \geq 0$ ,  $U' A - C \geq 0$  and  $X^* \geq 0$ , thus  $\pi + \lambda$  can be equal to

zero if and only if  $\pi$  and  $\lambda$  are both equal to zero. Thus, complementary slackness must hold at optimality.

The complementary slackness conditions are interpretable economically. The  $U'(AX^* - b) = 0$  condition implies that the resource will: a) be valued at zero if it is not fully utilized (i.e.,  $AX^* < b \implies U' = 0$ ) or b) have a nonzero value only if it is fully utilized (i.e.,  $AX^* = b \implies U' > 0$  [note a zero value could occur]). Thus, resources only take on value only when they have been exhausted. The condition  $(U' A - C)X^* = 0$  implies that a good will only be produced if its reduced cost is zero (i.e.,  $X^* > 0 \implies U' A - C = 0$ ) and that only zero  $X$ 's can have a reduced cost (i.e.,  $U' A - C > 0 \implies X^* = 0$ ). This last result also shows the returns ( $C$ ) to every nonzero variable are fully allocated to the shadow prices ( $U$ ) times the amount of resources ( $A$ ) used in producing that activity (i.e.,  $U' A = C$ ).

#### 4.2.4 Zero Profits

We have noted that  $U_i$  is the imputed marginal value of resource  $i$  and  $b_i$  is its endowment. Thus,  $U_i b_i$  is sometimes called the "payment" to resource  $i$ . When we sum over all  $m$  resources, the dual objective function can be interpreted as the total imputed value of the resource endowment. If the total imputed value of

the resources is viewed as a "cost", then it makes sense that firm should seek to find  $u_1, u_2, \dots, u_m$ , which minimizes  $\sum_i u_i b_i$ . However, at optimality the dual objective equals that of the primal

$$\sum_i u_i b_i = \sum_j c_j x_j.$$

Thus, total payments to the resources will equal the profit generated in the primal problem. All profits are allocated to resource values, and the solution insures that the imputed rate of the resources allocated by the primal problem are such that their total value equals total profits. Thus, if the firm had to pay  $U$  for the resources, zero profits would result.

#### 4.2.5 Finding the Dual Solution Information

When you have solved the dual, you have solved the primal. Thus given the optimal  $B^{-1}$ , the optimal dual variables are the primal shadow prices  $C_B B^{-1}$  without any need for solution. In general, one can show that the following correspondence holds (see Hadley (1962) or Bazaraa et al.).

Primal Solution Information	Corresponding Dual Solution Information
Objective function	Objective function
Shadow prices	Variable values
Slacks	Reduced costs
Variable values	Shadow prices
Reduced costs	Slacks

For example, if one wants to know the optimal values of the dual slacks, those values are the primal reduced costs.

### 4.3 Duality Under Other Model Forms

In the preceding discussion, the primal problem has always taken on standard form. We have seen that given a LP problem

$$\begin{aligned}
\text{Max} \quad & \sum_j c_j x_j \\
\text{s.t.} \quad & \sum_j a_{ij} x_j \leq b_i, \text{ for all } i \\
& x_j \geq 0, \text{ for all } j,
\end{aligned}$$

that its dual will always be

$$\begin{aligned}
\text{Min} \quad & \sum_i u_i b_i \\
\text{s.t.} \quad & \sum_i u_i a_{ij} \leq c_j, \text{ for all } j \\
& u_i \geq 0, \text{ for all } i.
\end{aligned}$$

Note that if the primal problem contains  $n$  variables and  $m$  constraints, the dual problem has  $m$  variables and  $n$  constraints. The dual for problems which are not in standard form can be written in two ways. One may convert a problem in non-standard form to reformulate it into standard form then write the dual, or one can learn primal-dual relationships for alternative primal problem forms. We discuss the second approach first.

The form of the primal constraints determines the restrictions on the sign of the associated dual variable. If the primal objective is to maximize, each  $\leq$  constraint has a corresponding non-negative ( $\geq 0$ ) dual variable. Each  $\geq$  constraint has a corresponding non-positive ( $\leq 0$ ) dual variable. Why? If a ( $\leq$ ) constraint is binding in a maximize primal, it follows that reducing the RHS of the constraint would make the constraint less binding and could only improve or leave unaffected the optimal objective function value. Thus, the objective function value is unchanged or decreases if the RHS of the constraint is increased and the associated dual variable is non-positive. An equality constraint in a primal problem gives a dual variable which is unrestricted in sign. The optimal solution to the primal problem must lie on the equality constraint. An

outward shift in the constraint could either increase or decrease the objective function, thus the corresponding dual variable is unrestricted in sign. These relationships are summarized in the first part of Table 4.1.

In regards to the primal variables, if the primal objective is to maximize, then each non-negative primal variable gives rise to a  $\leq$  constraint in the dual. If a primal variable is restricted to be non-positive, the corresponding dual constraint is a  $\geq$  inequality. Similarly, unrestricted primal variables lead to dual equalities. These results are summarized in the lower part of Table 4.1.

Table 4.1 may also be used to develop relationships for a minimization problem by reading the information from left to right. Suppose the objective is to minimize and a  $\geq$  constraint is present. The corresponding dual variable in a maximization dual would be non-positive.

A second approach can also be followed; i.e., always transform the problem to standard form then write its dual. To illustrate, consider the LP problem

$$\begin{aligned}
 \text{Max} \quad & 3x_1 + 2x_2 + x_3 \\
 \text{s.t.} \quad & x_1 + x_2 + 2x_3 \leq 20 \\
 & +2x_1 + x_2 + x_3 \leq 10 \\
 & x_1 \geq 0, \quad x_2 \leq 0, \quad x_3 \text{ unrestricted}
 \end{aligned}$$

Let us write the dual to this problem using the two approaches outlined above. First, let's convert the problem to standard form. To do this, the equality constraint must be replaced by two constraints:

$$\begin{aligned}
 & x_1 + x_2 + 2x_3 = 20 \\
 & + (x_1 + x_2 + 2x_3) \leq -20
 \end{aligned}$$

In addition the second primal constraint should be multiplied through by -1, the first variable is replaced by its negative ( $x_1 = -x_1'$ ) and the third variable  $x_3$  is replaced by  $x_3^+ - x_3^-$ . Making these substitutions and modifications gives

$$\begin{aligned}
\text{Max} \quad & 3x_1 + 2x_2 + x_3 \\
\text{s.t.} \quad & x_1 + x_2 + 2x_3 \leq 20 \\
& x_1 + x_2 + 2x_3 \leq 20 \\
& 2x_1 + x_2 + x_3 \leq 10 \\
& x_1, x_2, x_3 \geq 0
\end{aligned}$$

The dual to this problem is

$$\begin{aligned}
\text{Min} \quad & 20w_1 + 20w_2 + 10w_3 \\
\text{s.t.} \quad & w_1 + w_2 + 2w_3 \geq 3 \\
& w_1 + w_2 + w_3 \geq 2 \\
& 2w_1 + 2w_2 + w_3 \geq 1 \\
& 2w_1 + 2w_2 + w_3 \geq 1 \\
& w_1, w_2, w_3 \geq 0
\end{aligned}$$

Note that: a) the last two constraints can be rewritten as an equality, b) variables  $w_1$  and  $w_2$  can be combined into variable  $w_0 = w_1 - w_2$  which is unrestricted in sign, and c) we may substitute  $w_3^- = -w_3$  and we may revise the inequality of the first constraint yielding

$$\begin{aligned}
\text{Min} \quad & 20w_0 + 10w_3^- \\
\text{s.t.} \quad & w_0 + 2w_3^- \geq 3 \\
& w_0 + w_3^- \geq 2 \\
& 2w_0 + w_3^- = 1 \\
& w_0 \text{ unrestricted}, w_3^- \geq 0
\end{aligned}$$

which is the final dual.

This result can also be obtained from the use of the primal-dual relationships in Table 4.1. The primal objective is to maximize, so the dual objective is to minimize. The primal problem has 3 variables and 2



constraints, so the dual has 2 variables and 3 constraints. The first primal constraint is an equality, so the first dual variable is unrestricted in sign. The second primal constraint is a ( $\leq$ ) inequality so the second dual variable should be non-positive. The first primal variable is restricted to be non-positive, so the first dual constraint is a ( $\geq$ ) inequality  $x_2$  is restricted to be non-negative, thus the second dual constraint is  $\leq$ ;  $x_3$  is unrestricted in sign. Thus, the third dual constraint is an equality. Then, the dual can be written as

$$\begin{array}{llll}
 \text{Min} & 20u_1 & + & 10u_2 \\
 \text{s.t.} & u_1 & + & 2u_2 \leq 3 \\
 & u_1 & + & u_2 \leq 2 \\
 & 2u_1 & + & u_2 = 1 \\
 & u_1 \text{ unrestricted} & , & u_2 \geq 0
 \end{array}$$

which if one substitutes  $u_1 = w_0$  and  $u_2 = w_3$  is identical to that above.

#### 4.4 The Character of Dual Solutions

If the primal problem possesses a unique nondegenerate, optimal solution, then the optimal solution to the dual is unique. However, dual solutions arise under a number of other conditions. Several of the cases which can arise are:

- 1) When the primal problem has a degenerate optimal solution, then the dual has multiple optimal solutions.
- 2) When the primal problem has multiple optimal solutions, then the optimal dual solution is degenerate.
- 3) When the primal problem is unbounded, then the dual is infeasible.
- 4) When the primal problem is infeasible, then the dual is unbounded or infeasible.

#### 4.5 Degeneracy and Shadow Prices

The above interpretations for the dual variables depend upon whether the basis still exists after the change occurs. As shown in the previous chapter, there is a right hand side range over which the basis remains optimal. When a basic primal variable equals zero, the dual has alternative optimal solutions. The cause of this situation is generally that the primal constraints are redundant at the solution point and the range of right hand sides is zero. This redundancy means one does not need a full basic solution, so one of the basic variables is set to zero with the other basic variables likely to be nonzero. The best way to explain the implications of this situation is through an example. Consider the following problem

$$\begin{array}{rcll}
 \text{Max} & 3X_1 & + & 2X_2 \\
 & X_1 & + & X_2 & = & 100 \\
 & X_1 & & & = & 50 \\
 & & & X_2 & = & 50 \\
 & X_1 & , & X_2 & \geq & 0
 \end{array}$$

Notice that at the optimal solution,  $X_1 = 50$ ,  $X_2 = 50$ , the constraints are redundant. Namely, either the combination of the last two constraints or the first two constraints would yield the same optimal solution which is  $X_1 = X_2 = 50$ . The simplex solution of this problem shows a tie for the entering variable in the second pivot where one has the choice of placing  $X_2$  into the solution replacing either the slack variable from the first or the third constraint. If the first slack variable ( $S_1$ ) is chosen as basic then one gets  $X_1 = 50$ ,  $X_2 = 50$ ,  $S_1 = 0$  while  $S_1$  is basic. The associated shadow prices are 0, 3, and 2. On the other hand, if  $S_3$  were made basic one gets  $X_1 = 50$ ,  $X_2 = 50$ ,  $S_3 = 0$  with the shadow prices 2, 1, 0. Thus, there are two alternative sets of shadow prices, both of which are optimal. (Note, the dual objective function value is the same as the optimal primal in each case.)

The main difficulty with degeneracy is in interpreting the shadow price information. The shadow

prices are taking on a direction (i.e., see the arguments in McCarl (1977)). Note that if one were to increase the first right hand side from 100 to 101 this would lead to a zero change in the objective function and  $X_1$  and  $X_2$  would remain at 50. On the other hand if one were to decrease that right hand side from 100 to 99 then one would obtain an objective function which is two units smaller because  $X_2$  would need to be reduced from 50 to 49. This shows that the two alternative shadow prices for the first constraint (i.e., 0 and 2) each hold in a direction. Similarly if the constraint on  $X_1$  was increased to 51, the objective function increases by one dollar as one unit of  $X_2$  would be removed from the solution in favor of  $X_1$ ; whereas, if the constraint was moved downward to 49, it would cost three dollars because of the reduction in  $X_1$ . Meanwhile, reducing the constraint on  $X_2$  would cost two dollars, while increasing it would return to zero dollars. Thus in all three cases shadow prices take on a direction and the value of that change is revealed in one of the two dual solutions. This is quite common in degeneracy and may require one to do a study of the shadow prices or try to avoid degeneracy using a priori degeneracy resolution scheme as discussed in McCarl (1977); Paris (1991); Gal, and Gal et al., or as implemented automatically in OSL.

#### **4.6 Primal Columns are Dual Constraints**

One final comment relative to modeling is that the columns in the primal, form constraints on the dual shadow price information. Thus, for example, when a column is entered into a model indicating as much of a resource can be purchased at a fixed price as one wants, then this column forms an upper bound on the shadow price of that resource. Note that it would not be sensible to have a shadow price of that resource above the purchase price since one could purchase more of that resource. Similarly, allowing goods to be sold at a particular price without restriction provides a lower bound on the shadow price.

In general, the structure of the columns in a primal linear programming model should be examined to see what constraints they place upon the dual information. The linear programming modeling chapter extends

this discussion.

## References

- Bazaraa, M.S., J.J. Jarvis, and H.D. Sherali. Linear Programming and Network Flows. New York: John Wiley & Sons, 1990.
- Gal, T. "Shadow Prices and Sensitivity Analysis in Linear Programming under Degeneracy." OR Spektrum. 8(1986):59-71.
- Gal, T., H.J. Kruse, and P. Zörnig. "Survey of Solved and Open Problems in the Degeneracy Phenomenon." Mathematical Programming. 42(1988):125-133.
- Hadley, G. Linear Programming. Addison-Wesley Publishing Company, Inc., Reading, MA, 1962.
- McCarl, B.A. "Degeneracy, Duality and Shadow Prices in Linear Programming". Canadian Journal of Agricultural Economics. 25,1(1977):70-73.
- Paris, Q. An Economic Interpretation of Linear Programming. Iowa State University Press, Ames, Iowa, 1991.

**Table 4.1. Primal-Dual Correspondence for problems not in standard form**

Maximization Primal Problem	Minimization Dual Problem
<b>Form of Primal Constraints</b>	<b>Implication for Dual Variables</b>
$\sum_j a_{ij}x_j \leq b_i$	$u_i \geq 0$
$\sum_j a_{ij}x_j = b_i$	$u_i$ - unrestricted in sign
$\sum_j a_{ij}x_j \geq b_i$	$u_i \leq 0$
<b>Form of Primal Variables</b>	<b>Implication for Dual constraints</b>
$x_j \geq 0$	$\sum_i a_{ij}u_i \leq c_j$
$x_j$ - unrestricted in sign	$\sum_i a_{ij}u_i = c_j$
$x_j \leq 0$	$\sum_i a_{ij}u_i \geq c_j$

## CHAPTER V: LINEAR PROGRAMMING MODELING

In this chapter we concentrate heavily on the algebraic approach to LP modeling, GAMS usage, duality implications and some model specification issues. The chapter begins with a presentation of the general algebraic modeling approach with GAMS. Subsequently, we organize the presentation around commonly modeled situations. The first problem presented is the classical resource allocation problem. This is followed by the transportation and diet/feed mix/blending problems. Following these is a formulation that explicitly incorporates joint products. In presenting this material, we identify different types of variables and constraints used in building models, as well as examples of modeling assumptions used when formulating problems.

Applied duality is also treated. However, the duality material is not intended to imply that the dual of every problem must be formed when modeling. Instead, we discuss implications for the shadow prices that arise due to primal variable structures. The empirical relationship between primal and dual solutions will also be exhibited, which hopefully leaves no doubt that when solving the primal problem, the dual is simultaneously solved.

A third theme herein is communication of the empirical issues involved in LP application. This is attempted through the development of examples involving coefficient calculation and deductive steps in modeling. We will also demonstrate the link between algebraic representations of problems and empirical models. This discussion is designed to show readers the usefulness of algebraic models as a way of conceptualizing problems.

This particular chapter is not designed to stand alone. Additional formulations and algebraic/GAMS modeling techniques are presented throughout the rest of the book. Furthermore, reference to the GAMS manual (Brooke et al.) is essential to a thorough understanding of the material.

## 5.1 Introduction to Algebraic Modeling

Now we turn our attention to the construction of algebraic mathematical programming models using summation notation and GAMS. This section presents a brief introduction to a subset of GAMS. The GAMS manual (Brooke et al.) provides a complete treatment. This section is also supported by the material in Appendix A.

### 5.1.1 Defining Indices and GAMS Sets

A crucial algebraic modeling element is the identification of indices (also referred to mathematically as a subscript, or in GAMS as a SET). The compactness with which one can specify a model depends on the characterization of items with indices. However, the readability of the model depends on indices being disaggregate enough to reveal essential features. Examples of the appropriate specification of indices appear throughout the rest of the text.

Definition of indices in GAMS involves the definition of sets and set members. Namely, an index in summation notation is a SET in GAMS, and one specifies the range of the index by specifying SET membership. Examples of such set definitions are included in the following four GAMS statements.

SETS	I	THE INDEX I	/1,2,3,4,5/
	CITIES	CITY NAMES	/BOSTON, PORTLAND /
	YEAR	MODELS YEARS	/1970*1974/
	WEST (CITIES)	WESTERN CITIES	/PORTLAND/;

The first item entered is the label SETS which suggests that the following material contains SET definitions. Here four sets are being defined and are called I, CITIES, YEAR and WEST. The set WEST is defined as a subset of set CITIES. Within each set definition are three components; 1) the name of the set; 2) a verbal description of what the set stands for (which keeps a record for future users and helps describe the set in the output); and 3) a list of the set members enclosed between a pair of slashes with multiple set elements separated by commas (this can also contain a definition of the set element). When a set contains consecutively



numbered elements, an asterisk (as in the YEAR set definition) can be used to denote all elements in a numeric range (i.e., 1970\*1974 means include 1970, 1971, 1972, 1973, and 1974). Finally, note that a semicolon is used to end the SET statement after all sets have been defined. Set elements can be defined using numbers, characters, or a mixture of both.

In GAMS, one can define subsets. A subset such as WEST (CITIES) identifies selected elements identifying such things as the western cities. Brooke, et. al. present information on more complex specifications.

### 5.1.2 Writing Equations

The obvious purpose of algebraic modeling is to write general algebraically based equations which encompass a wide variety of situations. Two fundamental classes of algebraic equations exist. The first class encompasses all single equations which are not defined with respect to any kind of index. A scalar calculation falls into this class. The second class involves equations which are defined according to indices. This class involves families of simultaneous equations. For example, similar equations may be defined for each location, land class, and time period.

#### 5.1.2.1 Unindexed Equations

A summation notation version of an unindexed equation is

$$Z = \sum_j 3 a_j x_j$$

This equation depicts the sum across all the members of the index set j where three times a<sub>j</sub> is multiplied times x<sub>j</sub> and the resultant sum is placed into the value of Z. A related example is

$$Q = \sum_i \sum_j r_{ij}/y_{ij} \quad \text{No}$$

tice in both equations that all indices are included in the sum, and the result is a scalar number. This will always be true with unindexed or scalar equations. There cannot be indices or sets within the equation which are not summed over. The GAMS statements equivalent to the above two equations are

$$Z = \text{SUM}(J, 3 * A(J) * X(J));$$

$$Q = \text{SUM}((I,J), R(I,J) / Y(I,J));^2$$

The elements, of these GAMS equations, merit discussion. First, both equations equate a scalar to a sum. In doing this, the set or sets to be summed over are specified as the parameters to be used in forming the equation. Note, after the summation operator (SUM), the index or indices are named, followed by a comma, with the parameters separated by mathematical operators. Parentheses must be balanced. Finally, each expression is ended by a semicolon.

Although these examples are consistent with the algebraic models, they are poor examples of GAMS modeling. Namely GAMS formulations can be made much more self explanatory by utilizing set and set member names which are up to ten characters long. For example, the following two GAMS commands elaborate on those above, but are much easier to read;

$$\text{INCOME} = \text{SUM}(\text{PRODUCT}, 3 * \text{PRICE}(\text{PRODUCT}) * \text{QUANTITY}(\text{PRODUCT}));$$

$$\text{TOTALQUAN} = \text{SUM}((\text{SOURCE}, \text{DESTINATION}), \text{TOTALCOST}(\text{SOURCE}, \text{DESTINATION}) / \text{COSTPRUNIT}(\text{SOURCE}, \text{DESTINATION}));$$

Here the labels help define what the equation represents.

### 5.1.2.2 Indexed Equations

The other class of equations commonly used in algebraic modeling involves equations defined over a family of indices. Consider for example :

$$b_i = \sum_j a_{ij} X_j \quad \text{for all } i$$

$$r_{mj} = \sum_k (f_{km} Q_{kj} \% Z_{jm}) \quad \text{for all } m \text{ and } j$$

where the "for all" statement defines the family of indices for which the equations exist. Note that in these equations the index mentioned in the "for all" clause is not summed over. Rather, the "for all" clause indicates

---

<sup>2</sup> Note that these GAMS statements show equations for numerical calculations, not equations for inclusion as a constraint or objective function in a linear programming model. In such cases a slightly different syntax is utilized, as defined in the next section.

presence of these equations for each of these indices. The equivalent GAMS statements are

$$B(\text{ITEM}) = \text{SUM} (J, A(\text{ITEM},J)*X(J)) ;$$

$$R(\text{WHATISIT},J) = \text{SUM} (K, F(K,J,\text{WHATISIT})*Q(I,J)) + Z(J,\text{WHATISIT});$$

### 5.1.3 Data Definitions

When using algebraic modeling, data items are usually defined symbolically. For example,  $a_{ij}$  might be defined as the usage of the  $i^{\text{th}}$  resource per unit of the  $j^{\text{th}}$  product manufactured. In the development of an application, such items need to be defined numerically. In GAMS, raw data are entered using the SCALER, PARAMETER and TABLE syntax.

Suppose one wishes to enter single numbers which are not referenced with respect to a SET. In this case, one uses the SCALER syntax as follows:

```
SCALARS      LAND          LAND AVAILABLE      /100/
              HOURS        MACHINE TIME          /50/
              PRICESTEEL    STEEL PRICE              /2.25/;
```

Again the statements contain several parts. The keyword SCALER (or SCALARS) indicates unindexed data items are being defined. Then one enters each data item name followed by a verbal description and a numerical value enclosed in slashes. This can be followed by entries for as many scalars as necessary with the last definition followed by a semi-colon.

Data are also entered using the PARAMETER syntax. Parameter statements are used to enter indexed (subscripted) data in list form. Examples are given:

```
PARAMETERS
PRICE (ITEM)      ITEM PRICES          /LABOR 4.00, COAL 100/
DISTANCE (I,J)    INTER CITY DISTANCES /BOSTON.CHICAGO 20,
                                                         BOSTON.CLEVELAND 15/;
```

This syntax must begin with the word PARAMETER (or PARAMETERS). Subsequently, the name of the data item, its indices description appear followed by a slash. Within the slashes the name of each data

set index and their values are given. Commas separate set members and their values. This sequence repeats for each set member to be specified. In the examples, LABOR and COAL are members of the set ITEM with their associated entries in the PRICE data taking on the values 4.00 and 100 respectively. Similarly, the set I includes BOSTON and the set J includes CHICAGO and CLEVELAND. Note, the syntax in the second expression where BOSTON.CHICAGO indicates this is the data for a pair of elements (i value then j value). Finally, a semicolon indicates the PARAMETER statement end.

Yet another data entry possibility involves the TABLE syntax. This syntax is most convenient for defining multi-dimensional parameters as illustrated by:

TABLE DISTANCE(I,J) DISTANCE BETWEEN CITIES		
	CHICAGO	CLEVE
BOSTON	20	10
BALT	20	9

;

TABLE MODECOST(I,J,MODE) INTER CITY COST OF MOVING ITEMS

	TRUCK	RAIL
BOSTON.CHICAGO	10	5
BOSTON.CLEVE	8	7
BALT.CHICAGO	10	5
BALT.CLEVE	7	6

;

or

TABLE MODECOST(I,J,MODE) INTER CITY COST OF MOVING ITEMS

	CHICAGO.TRUCK	CHICAGO.RAIL	CLEVE.TRUCK	CLEVE.RAIL
BOSTON	10	5	8	7
BALT	10	5	7	6

;

The sets appear in the table structure in the order they are listed. Thus, for the above table the two dimensional item DISTANCE (I,J) is defined over the sets I and J with the names of the members of the set I appearing vertically down the left-hand side of the table while the member names for J are identified

horizontally. Data values appear underneath the names of the members of the J set.

The second and third example present alternative ways of entering a set with more than two dimensions. In the first case the first two sets are handled in the vertical column, and in the second case the last two sets are handled across the top. In both cases, the two sets are handled by identifying the member of the first indexed set followed by a period then the member name from the second set.

#### 5.1.4 Entering Data Through Calculation

One may also calculate data. In this case, the set dependence of a parameter and a verbal description is specified without data. Then an equation is specified to define the data. For example, suppose the cost of transporting items was a known linear function of mode and distance. A procedure to implement this would be as follows:

```
PARAMETER MODECOST (I,J, MODE)      COMPUTED COST OF SHIPPING BY MODE;
TABLE COSTUNIT (TERM, MODE)        COST DATA FOR SHIPPING MODE
                                     TRUCK   RAIL
                                     -----
                                     CONS    1     1.2
                                     PERMILE .5    .25   ;

MODECOST (I, J, MODE) = COSTUNIT ("CONS",MODE) + COSTUNIT ("PER
MILE",MODE)*DISTANCE (I,J);
DISPLAY MODECOST;
```

In this example the PARAMETER statement does not contain data. Rather, calculations define the variable MODECOST based on data in the tables DIST and COSTUNIT from the previous page which give distance and the dependence of transport cost on distance. After MODECOST is calculated, it is copied to the output using a DISPLAY statement. The data calculation feature is powerful as one can enter raw data and then do calculations of model parameters, and record the assumptions made in calculating the data.

One also can do simple replacements. For example, the statement

```
MODECOST(I,J,MODE) = 5.;
```

would result in having all elements of MODECOST equal to 5.

### 5.1.5 Specification of Variables

In linear programming models, the variables will have nonnegativity or other sign conditions. In GAMS this is identified by denoting POSITIVE VARIABLES, NEGATIVE VARIABLES, VARIABLES which are free variables, and later BINARY or INTEGER VARIABLES. For example

POSITIVE VARIABLES	
VOLUME (ITEM)	AMOUNT PRODUCED BY ITEM
QUANTITY (CITY1, CITY2)	AMOUNT SHIPPED BETWEEN CITIES;
NEGATIVE VARIABLES	
LOSS	AMOUNT OF MONEY LOST;
VARIABLES	
OBJ	OBJECTIVE FUNCTION VALUE;

Note that the variable and its sign restriction are simultaneously defined. A POSITIVE specification means that the variable is restricted to be greater than or equal to zero, NEGATIVE means less than or equal to zero, and the word VARIABLE means unrestricted in sign. After these statements, the name of each variable along with its set dependence appears, and may be followed by a short description of the variable. The statement lists each variable with the particular sign restriction, followed by a semicolon. Every variable used in a GAMS model must be defined and the model must contain at least one unrestricted variable which is the item optimized.

### 5.1.6 Equations

The set of equations which appear in the model are specified through the EQUATIONS statement.

EQUATIONS	
OBJECTIVE	OBJECTIVE FUNCTION
AVAILABLE (RESOURCES)	RESOURCE AVAILABILITY CONSTRAINTS
SUPPLY (PLANT)	SUPPLY AVAILABLE AT A PLANT
LIMIT (RESOURCES, PLANT)	RESOURCES AVAILABLE BY PLANT;

These expressions include the name of the equation, an indication of the sets for which this equation

exists and a verbal description. Entries appear for each of the model equations followed by a semicolon.

### 5.1.7 Writing Model Equations

After the model equations have been defined, then they are defined algebraically. The algebraic specification differs somewhat depending on whether the objective function or the constraints are being entered.

#### 5.1.7.1 Objective Function Equations

The objective function equation is typically of the form:

$$\text{Max } \sum_j c_j x_j$$

However, GAMS forces the modeler to rewrite the equation so that the objective function expression is set equal to an unrestricted variable, and that variable is named in the solve statement. Thus we alter the problem to become

$$\begin{array}{ll} \text{Max} & Z \\ \text{S.t.} & Z = \sum_j c_j x_j \end{array}$$

where Z is a variable which is unrestricted in sign (named in the GAMS VARIABLES list).

Specifying the objective function equation in GAMS requires the specification of an equation name followed by an algebraic representation. Suppose we label the objective function equation OBJECTIVEF.

The resultant GAMS statement is :

$$\text{Objectivef.. } Z =E= \text{SUM}(J, c(J)*X(J));$$

Note the structure, first the equation is named followed by two periods. Then the algebraic statement appears where the unrestricted variable that is maximized is set equal to an expression for the objective function following the rules for equation writing as discussed above with one exception. That is, here the

equality is written =E= . Finally, the expression is followed by a semicolon.

### 5.1.7.2 Constraint Equations

The constraint equations are entered in essentially the same form as the objective function. First the equation is named. In this naming any indices(sets) over which the equations are defined are named. Then this is followed by two periods and the algebraic equation for the constraint. Subsequently, an indication of the form of the inequality appears (=L= for less than or equal to; =G= for greater than or equal to; and =E= for equal to). Finally, this is followed by the expression for the right-hand-side and a semi-colon. Consider for example the constraints:

$$\begin{aligned} & \sum_j a_{ij} X_j \# b_i \quad \text{for all } i \\ & \sum_k Z_k \$ d_j \quad \text{for all } j \\ & \sum_j \left( \sum_i (f_{ijm} Q_{ij}) \% Z_{jm} \right) ' 4 \quad \text{for all } m \\ & \sum_m \left( \sum_j \left( \sum_i (f_{ijm} Q_{ij}) \% Z_{jm} \right) \right) ' 0 \end{aligned}$$

where the "for all" statement defines the family of indices for which the equations exist. Suppose we label the first constraint RESOURCE indexed over the set ITEM. Similarly, we call the second constraint DEMAND indexed over PLACE, the third constraint SOMEITEM indexed over WHAT and the fourth ADDIT. The resultant GAMS statements are:

```
RESOURCE(ITEM).. SUM (J, A(ITEM,J)*X(J)) =L= B(ITEM);
DEMAND(PLACE).. SUM (K, Z(K,PLACE)) =G= D(PLACE);
someitem(what).. Sum(j,sum(i,f(i,j,what)*q(i,j))+z(j,what)) =e=4;
addit.. Sum((m,j),sum(i,f(i,j,what)*q(i,j))+z(j,what)) =e=0;
```

The equations follow the equation writing conventions discussed above and in Appendix A. The only exception is that one may mix constants and variables on either side of the equation. Then one could write the



first equation as:

$$\text{RESOURCE}(\text{ITEM}) - B(\text{ITEM}) = G = \text{SUM} (J, A(\text{ITEM},J)*X(J));$$

### 5.1.8 Specifying Bounds

Upper and lower bounds on individual variables are handled in a different fashion. For example, if the variables VOLUME and QUANTITY had been defined, and were to be bounded, then the bounds would be defined through statements like the following

```
VOLUME.UP (ITEM) = 10.;
QUANTITY.LO (CITY1, CITY2) = MINQ (CITY1, CITY2);
VOLUME.UP ("TREES") = 200. + LIMIT ("TREES").
```

In the first example, upper bounds are specified for the variable VOLUME across all members of the set ITEM equaling 10. In the second example, the bound is imposed through a prespecified parameter data. In the third example, the bound is calculated.

### 5.1.9 Forming and Solving a Model

After specifying sets, variables, data, and other appropriate input, one enters statements which define the model and invoke the solver. The MODEL definition can be of two forms. In the first form, the expression

```
MODEL RESOURCE /ALL/ ;
```

gives the model a name (RESOURCE), and specifies that ALL equations are incorporated. Alternatively, the statement

```
MODEL COSTMIN /OBJECTIVE,SUPPLY,DEMAND/;
```

gives the model name and the names of the equations to include. In this case the model name is COSTMIN and the equations included are OBJECTIVE, SUPPLY and DEMAND.

In turn, the SOLVE statement is

```
SOLVE COSTMIN USING LP MINIMIZING OBJ;
```

where the general syntax is SOLVE "modelname" USING LP MAXIMIZING [OR MINIMIZING] "objective function variable name." The objective function variable name is the variable added and set equal to the mathematical expression for the objective function one may also solve mixed integer problems by altering the using phrase to "usingmip" and nonlinear problems with "using NLP".

### 5.1.10 Report Writing

GAMS has useful features which allow solution results to be used in calculations to compute various items of interest or do various result summaries. One can, at any time after the SOLVE statement, use: a) a variable name followed by a .L to obtain the value of the solution for that variable; b) a variable name followed by a .M to get the reduced cost associated with that variable; c) an equation name followed by .L to get the value of the left hand side of that equation; and d) an equation name followed by .M to get the shadow price associated with the equation. Slack variables can be computed by taking an equation right hand side minus the associated .L value. The DISPLAY statement can be used to print out calculation results.

### 5.1.11 Additional GAMS Topics

Two additional GAMS topics are worth mentioning. These involve the use of conditional statements and the use of loops. Conditional statements are utilized to indicate that there are cases where particular items should not be defined when doing a calculation and involve a syntax form using a \$.

Two examples are

```
DEMAND(REGION)$DEMANDQ(REGION)..  
SUM (SUPPLY, QUAN(SUPPLY, REGION)) =G= DEMANDQ(REGION);  
OBJECTIVE.. OBJ = SUM((I,J)$ (DIST(I,J) LT 20), COST(I,J)*QUAN(I,J));
```

In the first case, the notation \$DEMANDQ(REGION) tells GAMS to generate this equation only if the DEMANDQ term is nonzero resulting in a DEMAND equation being generated for only those members of REGION which have nonzero demand. In the second expression, the \$(DIST(I,J) LT 20) clause indicates that one should add (i,j) pairs into the sum only if the DISTANCE between the pair is less than 20. In general, use

of the \$ notation allows one to impose conditions within the model setup and data calculation procedures.

LOOP statements are also worth mentioning as they can be used to repeatedly execute a set of GAMS statements. Examples appear in the DIET example on the disc where the LOOP is utilized to vary a price and solve over and over again. Similarly, the EVPORTFO example uses a LOOP to solve for alternative risk aversion parameters.

## **5.2 Putting GAMS and Algebraic Modeling Together -- The Resource Allocation Problem**

The classical LP problem involves the allocation of an endowment of scarce resources among a number of competing products so as to maximize profits. In formulating this problem algebraically let us pursue the logical steps of: a) index identification, b) variable identification, c) constraint identification and d) objective function identification. Parameter identification is implicit in steps c and d.

The indices necessary to formulate this problem are inherent in the problem statement. Since there are "competing products" we need an index for products which we define here as  $j$ . The index  $j$  may take on the values of 1 through  $n$  indicating that there are  $n$  different competing products. The problem statement also mentions "scarce resources", so we define the index  $i$  to represent each of  $m$  possible resources ( $i=1,2,\dots,m$ ).

Now suppose we turn our attention to variables. Two potential variables are implicit in the above statement. The first type involves variables indicating the allocation of resources and the second type is associated with the competing products. In structuring this problem we would have to know the exact amount of each resource that is used in making each product. Thus, when we know how much of a product is made, we know the quantity of each resource allocated to that product. This means the variable in this problem involves the amount of each product to be made. Suppose we define a variable  $X_j$  which is the number of units of the  $j^{\text{th}}$  product made.

We may now define constraint equations. In this case, a constraint is needed for each of the scarce resources. The constraint forces resource usage across all production possibilities to be less than or equal to

the resource endowment. An algebraic statement of this restriction requires definitions of parameters for resource usage by product and the resource endowment. Let  $a_{ij}$  depict the number of units of the  $i^{\text{th}}$  resource used when producing one unit of the  $j^{\text{th}}$  product and let  $b_i$  depict the endowment of the  $i^{\text{th}}$  resource. We now can construct an algebraic form of the constraint. Namely we need to sum up the total usage of the  $i^{\text{th}}$  resource. The total usage of resource is the sum of the per unit resource usage times the number of units produced ( $a_{ij} X_j$ ). The appropriate algebraic statement of the  $i$ th constraint is

$$\sum_j a_{ij} X_j \leq b_i,$$

with a constraint defined for each of the resources ( $i$ ) in the problem. A condition is also needed which states only zero or positive production levels are allowed for each production possibility.

$$X_j \geq 0, \text{ for all } j.$$

Turning to the objective function, an equation for profits earned is also needed. This involves the definition of a parameter for the profit contribution per unit of the  $j^{\text{th}}$  product ( $c_j$ ). Then, the algebraic statement of the objective function is the sum of the per unit profit contribution times the amount of each product made or

$$\sum_j c_j X_j$$

which is to be maximized.

Algebraically, the LP formulation of the problem is

$$\begin{aligned} \text{Max} \quad & \sum_j c_j X_j \\ \text{s.t.} \quad & \sum_j a_{ij} X_j \leq b_i \text{ for all } i \\ & X_j \geq 0 \text{ for all } j \end{aligned}$$

where  $j$  is indexed from 1 to  $n$ ,  $i$  is indexed from 1 to  $m$ ,  $c_j$  is the profit per unit of activity  $j$ ,  $X_j$  is the number of units of activity  $j$  produced,  $a_{ij}$  is the number of units of resource  $i$  required per unit of product  $j$ , and  $b_i$  is the endowment of resource  $i$ .

This formulation arose early in the development of LP. While the exact problem is not stated as above, there are problems very close to it in Koopman's 1949 symposia report. The formulation explicitly appears in Dorfman's 1951 book. Kantorovich also presents an early example. Over time, this particular formulation has been widely used. In fact, it provides the first example in virtually all textbooks.

### 5.2.1 Example and GAMS Implementation

Suppose that E-Z Chair Makers are trying to determine how many of each of two types of chairs to produce. Further, suppose that E-Z has four categories of resources which constrain production. These involve the availability of three types of machines: 1) large lathe, 2) small lathe, and 3) chair bottom carver; as well as labor availability. Two types of chairs are produced: functional and fancy. A functional chair costs \$15 in basic materials and a fancy chair \$25. A finished functional chair sells for \$82 and a fancy chair for \$105. The resource requirements for each product are shown in Table 5.1.

The shop has flexibility in the usage of equipment. The chairs may have part of their work substituted between lathes. Labor and material costs are also affected. Data on the substitution possibilities are given in Table 5.2. Assume the availability of time is 140 hours for the small lathe, 90 hours for the large lathe, 120 hours for the chair bottom carver, and 125 hours of labor.

This problem can be cast in the format of the resource allocation problem. Six different chair/production method possibilities can be delineated. Let

$X_1$  = the number of functional chairs made with the normal pattern;

$X_2$  = the number of functional chairs made with maximum use of the small lathe;

$X_3$  = the number of functional chairs made with maximum use of the large lathe;

$X_4$  = the number of fancy chairs made with the normal pattern;

$X_5$  = the number of fancy chairs made with maximum use of the small lathe;

$X_6$  = the number of fancy chairs made with maximum use of the large lathe.

The objective function coefficients require calculation. The basic formula is that profits for the  $j^{\text{th}}$  activity ( $c_j$ ) equal the revenue to the particular type of chair less the relevant base material costs, less any relevant cost increase due to lathe shifts. Thus, the  $c_j$  for  $X_1$  is calculated by subtracting 15 from 82, yielding the entered 67. The constraints on the problem impose the availability of each of the four resources. The technical coefficients are those appearing in Tables 5.1 and 5.2.

The resultant LP model in algebraic form is

$$\begin{array}{rllllllll}
 \text{Max} & 67X_1 & + & 66X_2 & + & 66.3X_3 & + & 80X_4 & + & 78.5X_5 & + & 78.4X_6 \\
 \text{s.t.} & 0.8X_1 & + & 1.3X_2 & + & 0.2X_3 & + & 1.2X_4 & + & 1.7X_5 & + & 0.5X_6 & \# & 140 \\
 & 0.5X_1 & + & 0.2X_2 & + & 1.3X_3 & + & 0.7X_4 & + & 0.3X_5 & + & 1.5X_6 & \# & 90 \\
 & 0.4X_1 & + & 0.4X_2 & + & 0.4X_3 & + & X_4 & + & X_5 & + & X_6 & \# & 120 \\
 & X_1 & + & 1.05X_2 & + & 1.1X_3 & + & 0.8X_4 & + & 0.82X_5 & + & 0.84X_6 & \# & 125 \\
 & X_1 & , & X_2 & , & X_3 & , & X_4 & , & X_5 & , & X_6 & \$ & 0
 \end{array}$$

This problem can be cast into an algebraic modeling system like GAMS in numerous ways. Two approaches are presented here. The first is faithful to the above algebraic formulation and the second is more tailored to the problem. Consider the first formulation as shown in Table 5.3 and the file called RESOURCE. The approach we employ in setting up the GAMS formulation involves a more extensive definition of both variables and constraints than in the algebraic format. The indices are defined as GAMS sets, but instead of using the short names  $i$  and  $j$  longer names are used. Thus, in statement one the set  $j$  (which stands for the different products that can be produced) is named PROCESS and the definition of the elements of  $j$  (rather than being 1-6) are mnemonics indicating which of the products is being produced and which lathe is being used. Thus, FUNCTNORM stands for producing functional chairs using the normal process while the mnemonic FANCYMLRG stands for producing fancy chairs with the maximum use of the large lathe. Similarly, in line four, the  $i$  subscript is named RESOURCE. In turn, the parameters reflect these definitions.

The availability of each resource ( $b_i$ ) is specified through the RESORAVAIL parameter in lines 13-15. The per unit use of each resource by each production process ( $a_{ij}$ ) is entered through the RESOURSUSE definition in lines 17-28. The objective function profit parameter ( $c_j$ ) is not directly defined but rather is calculated during model setup. The inputs to this calculation are defined in the data with the per unit prices of the chairs defined by process in PRICE lines 8-9 and the production cost defined in lines 11-12. In turn, the parameters are used in computing the objective function in line 39 where price minus production cost is computed.

The variables ( $X_j$ ) are defined in lines 30-31 where the POSITIVE specification means greater than or equal to zero. Note that a variable named PROFIT is defined in line 33. This variable does not have a counterpart in the algebraic model. This unrestricted variable is required by GAMS and will be the quantity maximized. Subsequently, the two equations are specified. The equation for the objective function is not indexed by any sets and is defined in line 35. The resource constraint equation indexed by the set RESOURCE appears in line 36. In turn then, the algebraic statement of the objective function appears in lines 38-40 which sums net profits over all possible production processes by computing price minus the production cost times the level produced. The resource constraints sum resource use over all the processes and hold this less than or equal to the resource availability (lines 42-44). The final step in the GAMS implementation is to specify the model name which contains all the equations (line 46) and then a request to solve the model (line 47).

Another GAMS formulation of the problem appears in Table 5.4 and file RESOURC1. This is less faithful to the algebraic formulation, but is presented here since it shows more about GAMS. Here, the  $j$  subscript definition is broken into two parts. One of which reflects the type of CHAIR being made and the second the type of PROCESS being utilized. Thus, FUNCT.NORMAL refers to a functional chair using the normal process. In turn, this allows PRICE and BASECOST to be specified by CHAIR, but the EXTRACOST from using the additional processes needs to be specified in terms of both CHAIR and

PROCESS. The resource usage matrix now has three dimensions: one for RESOURCE, one for CHAIR, and one for PROCESS. Also, the dimensions of the RESOURCUSE array are changed accordingly (line 20). The model then proceeds basically as before with the PRODUCTION variable now having two indices, CHAIR and PROCESS (line 33). The specification also means a slightly more complex expression in the objective function in which the net return to a particular chair, using a particular process is calculated as chair price minus production cost, minus extra cost. In turn, that is multiplied by the production level and summed (lines 41-42).

Examining and contrasting these formulations shows some of the power of GAMS. Namely, in both formulations, one can put in raw information such as chair prices and costs and then computations to setup the model. This leaves a record of how the numbers were calculated rather than having model coefficients computed elsewhere. In addition, the use of longer names (up to 10 characters) in specifying the model, means that the GAMS instructions can be written in sufficient detail so that they are easily understood. For example, look at the objective function equation in Table 5.4. Note, that it contains production level ties, the price of the chair minus the cost of the chair minus the extra cost by a process. This is a much more readily accessible formulation than exists in many computational approaches to linear programming (see the material below or treatments like that in Beneke and Winterboer).

### **5.2.2 Model Solution**

The resultant solution under either formulation gives an optimal objective function value of 10,417.29. The optimal values of the primal variables and shadow prices are shown in Table 5.5. This solution indicates that 62.23 functional chairs and 73.02 fancy chairs should be made using the normal process, while 5.18 fancy chairs should be made using maximum use of a large lathe. This production plan exhausts small and large lathe resources as well as labor. The dual information indicates that one more hour of the small lathe is worth \$33.33, one more hour of the large lathe \$25.79, and one more hour of labor \$27.44. The reduced cost



valuation information also shows, for example, that functional chair production with maximum use of a small lathe would cost \$11.30 a chair. Finally, there is excess capacity of 16.91 hours of chair bottom carving.

### 5.2.3 Comments

The resource allocation problem is the most fundamental LP problem. This problem has many uses; however, most uses involve slightly more complex problem structures which will be discussed in the remainder of the book.

A number of modeling assumptions are implicit in this formulation. First, the price received for production of chairs is independent of the quantity of chairs produced. The firm would receive the same price per unit whether it produced 5 or 500 million chairs. This may be unrealistic for large firms. Increased production may bring either an increasing or decreasing price. Representation of decreasing returns is presented in the nonlinear approximation chapter and in the price endogenous modeling chapter. The integer programming chapter contains a formulation for the case where prices increase with sales.

A second assumption of the above formulation is that the fixed resource availability does not change regardless of its value. For example, the E-Z chair problem solution placed a value on labor of \$27 an hour. However, the firm may feel it can afford more labor at that price. Consequently, one may wish to extend the model so that more of the resources become available if a sufficiently high price would be paid. This general topic will be covered under purchase activities which are introduced in the joint products problem below. It is also covered in the nonlinear approximation and price endogenous chapters.

Finally, consider an assumption that does not characterize the above formulation. Many people ordinarily regard the resource allocation problem as containing a fixed coefficient production process where there is only a single way of producing any particular product. However, we included multiple ways of producing a product in this example problem to show that a LP model may represent not only one way of

producing a product, but also different ways involving different combinations of inputs.

### 5.3 The Transportation Problem

The second problem covered is the transportation problem. This problem involves the shipment of a homogeneous product from a number of supply locations to a number of demand locations. This problem was originally formulated by Kantorovich in 1939 and Hitchcock in 1941. Koopmans (1949) restated the model and spurred research in the area.

Setting this problem up algebraically requires definition of indices for: a) the supply points which we will designate as  $i$ , and b) the demand locations which we will designate as  $j$ . In turn, the variables indicate the quantity shipped from each supply location to each demand location. We define this set of variables as  $X_{ij}$ .

There are three general types of constraints, one allowing only nonnegative shipments, one limiting shipments from each supply point to existing supply and the third imposing a minimum demand requirement at each demand location. Definition of the supply constraint requires specification of the parameter  $s_i$  which gives the supply available at point  $i$ , as well as the formation of an expression requiring the sum of outgoing shipments from the  $i^{\text{th}}$  supply point to all possible destinations ( $j$ ) to not exceed  $s_i$ . Algebraically this expression is

$$\sum_j X_{ij} \leq s_i.$$

Definition of the demand constraint requires specification of the demand quantity  $d_j$  required at demand point  $j$ , as well as the formation of an expression summing incoming shipments to the  $j^{\text{th}}$  demand point from all possible supply points ( $i$ ). Algebraically this yields

$$\sum_i X_{ij} \geq d_j.$$

Finally, the objective function depicts minimization of total cost across all possible shipment routes. This involves definition of a parameter  $c_{ij}$  which depicts the cost of shipping one unit from supply point  $i$  to demand point  $j$ . In turn, the algebraic formulation of the objective function is the first equation in the composite formulation below.

$$\begin{aligned}
 \text{Min} \quad & \sum_i \sum_j c_{ij} X_{ij} \\
 \text{s.t.} \quad & \sum_j X_{ij} \leq s_i \quad \text{for all } i \\
 & \sum_i X_{ij} \geq d_j \quad \text{for all } j \\
 & X_{ij} \geq 0 \quad \text{for all } i, j
 \end{aligned}$$

This particular problem is a cost minimization problem rather than a profit maximization problem. The transportation variables ( $X_{ij}$ ) belong to the general class of transformation variables. Such variables transform the characteristics of a good in terms of form, time, and/or place characteristics. In this case, the transportation variables transform the place characteristics of the good by transporting it from one location to another. The supply constraints are classical resource availability constraints. However the demand constraint imposes a minimum level and constitutes a minimum requirement constraint.

Suppose we address duality before turning to GAMS and an example. The dual of the transportation problem establishes imputed values (shadow prices) for supply at the shipping points and demand at the consumption points. These values are established so that the difference between the value of demand and the cost of supply is maximized. The dual problem is given by<sup>3</sup>

---

<sup>3</sup> This formulation follows after the supply equation has been multiplied through by -1 to transform it to a greater-than constraint.

$$\begin{aligned}
 \text{Max } & \sum_i s_i U_i - \sum_j d_j V_j \\
 \text{s.t. } & U_i - V_j \leq c_{ij} \text{ for all } i, j \\
 & U_i, V_j \geq 0 \text{ for all } i, j
 \end{aligned}$$

where  $U_i$  is the marginal value of one unit available at supply point  $i$ , and  $V_j$  is the marginal value of one unit demanded at demand point  $j$ .

The dual problem is best explained by first examining the constraints. The constraint associated with  $X_{ij}$  requires the value at the demand point ( $V_j$ ) to be less than or equal to the cost of moving the good from point  $i$  to point  $j$  ( $c_{ij}$ ) plus the value at the supply point ( $U_i$ ). Consequently, the model requires the marginal value of supply at the supply point plus the transportation cost to be no smaller than the value at the demand point. This also requires the differences in the shadow prices between demand and supply points to be no greater than the transport cost. This requirement would arise in a highly competitive market. Namely if someone went into the transportation business quoting delivery and product acquisition prices, one could not charge the demanders of the good more than the price paid for a good plus the cost of moving it, or competitors would enter the market taking away business. This also shows the general nature of the dual constraint imposed by a primal transformation variable. Namely, a restriction is imposed on the difference in the shadow prices. The dual objective function maximizes profits which equal the difference between the value of the product at the demand points ( $d_j V_j$ ) and the cost at the supply points ( $s_i U_i$ ).

### 5.3.1 Example

ABC Company has three plants which serve four demand markets. The plants are in New York, Chicago and Los Angeles. The demand markets are in Miami, Houston, Minneapolis and Portland. The

quantity available at each supply point and the quantity required at each demand market are

<u>Supply Available</u>		<u>Demand Required</u>	
New York	100	Miami	30
Chicago	75	Houston	75
Los Angeles	90	Minneapolis	90
		Portland	50

The assumed distances between cities are

	Miami	Houston	Minneapolis	Portland
New York	3	7	6	23
Chicago	9	11	3	13
Los Angeles	17	6	13	7

Assume that the firm has discovered that the cost of moving goods is related to distance (D) by the formula --  $\text{Cost} = 5 + 5D$ . Given these distances, the transportation costs are

	Miami	Houston	Minneapolis	Portland
New York	20	40	35	120
Chicago	50	60	20	70
Los Angeles	90	35	70	40

The above data allow formulation of an LP transportation problem. Let  $i$  denote the supply points where  $i=1$  denotes New York,  $i=2$  Chicago, and  $i=3$  Los Angeles. Let  $j$  represent the demand points where  $j=1$  denotes Miami,  $j=2$  Houston,  $j=3$  Minneapolis, and  $j=4$  Portland. Next define  $X_{ij}$  as the quantity shipped from city  $i$  to city  $j$ ; e.g.,  $X_{23}$  stands for the quantity shipped from Chicago to Minneapolis. A formulation of

this problem is given in Table 5.6. This formulation may also be presented in a more compact format as

20	40	35	120	50	60	20	70	90	35	70	40	Minimize	
1	1	1	1									#	100
				1	1	1	1					#	75
								1	1	1	1	#	90
+1				+1				+1				\$	30
	+1				+1				+1			\$	75
		+1				+1				+1		\$	90
			+1				+1				+1	\$	50

This shows the common incidence in LP formulations of sparsity. Although there is room for 84 coefficients in the body of the constraint matrix only 24 of these are non-zero. Thus, the problem is only 24/84ths dense, revealing a large number of zeros within the body of the matrix. The right hand sides illustrate endowments of supply through the first three constraints, and minimum requirements in the next four constraints. The variables involve resource usage at the supply point and resource supply at the demand point. These activities transform the place utility of the good from the supply point to the demand point.

The GAMS implementation is presented in Table 5.7 and in the file TRANSPORT. Two sets are defined (lines 1-4) one identifies the supply plants, the other defines the demand markets. Subsequently, supply availability and demand requirements are specified through the parameter statements in lines 6-10. The distance between the plants is specified in lines 12-17. Following this, unit transport costs are computed by first defining the parameter for cost in line 20, and then expressing the formula of \$5.00 fixed cost plus \$5.00 times the distance in line 21. Next, the nonnegative shipment variables are specified ( $X_{ij}$ ). The unrestricted variable called TCOST equals the objective function value. Then, three equations are specified, one equates the objective value to TCOST and the other two impose the supply and demand limitations as in the algebraic model. These in turn are specified in the next few statements, then the model is specified using all the

equations and solved minimizing the TCOST variable.

The solution to this problem is shown in Table 5.8. The optimal value of the objective function value is 7,425. The optimal shipping pattern is shown in Table 5.9. The solution shows twenty units are left in New York's potential supply (since constraint 1 is in slack). All units from Chicago are exhausted and the marginal value of additional units in Chicago equals \$15 (which is the savings realized if more supply were available at Chicago which allowed an increase in the volume of Chicago shipments to Minneapolis and thereby reducing New York-Minneapolis shipments).

The solution also shows what happens if unused shipping routes are used. For example, the anticipated increase in cost that would be necessary if one were to use the route from New York to Portland is \$75, which would indicate a reshuffling of supply. For example, Los Angeles would reduce its shipping to Portland and increase shipping to somewhere else (probably Houston).

The associated dual problem is:

$$\begin{array}{rllllllll}
 \text{Max} & -100U_1 & - & 75U_2 & - & 90U_3 & + & 30V_1 & + & 75V_2 & + & 90V_3 & + & 50V_4 \\
 \text{s.t.} & -U_1 & & & & & + & V_1 & & & & & & \# & 20 \\
 & -U_1 & & & & & & & + & V_2 & & & & \# & 40 \\
 & -U_1 & & & & & & & & & + & V_3 & & \# & 35 \\
 & -U_1 & & & & & & & & & & + & V_4 & \# & 120 \\
 & & & -U_2 & & & + & V_1 & & & & & & \# & 50 \\
 & & & -U_2 & & & & & + & V_2 & & & & \# & 60 \\
 & & & -U_2 & & & & & & & + & V_3 & & \# & 20 \\
 & & & -U_2 & & & & & & & & + & V_4 & \# & 70 \\
 & & & & & -U_3 & + & V_1 & & & & & & \# & 90 \\
 & & & & & -U_3 & & & + & V_2 & & & & \# & 35 \\
 & & & & & -U_3 & & & & & + & V_3 & & \# & 70 \\
 & & & & & -U_3 & & & & & & + & V_4 & \# & 40
 \end{array}$$

The constraints of this problem are written in the less than or equal to, for requiring the value at demand point to be no more than the value at the shipping point plus the transportation cost. The solution to the dual problem yields an objective function of 7,425 with the optimal values for the variables shown in Table 5.10.

A comparison of Table 5.8 and Table 5.10 reveals the symmetry of the primal and dual solutions. The values of the  $X_{ij}$  in the primal equal the shadow prices of the dual, and vice versa for the shadow prices of the primal and the  $U_i$  and  $V_j$  in the dual.

### 5.3.2 Comments

The transportation problem is a basic component of many LP problems. It has been extended in many ways and has been widely used in applied work. A number of assumptions are contained in the above model. First, transportation costs are assumed to be known and independent of volume. Second, supply and demand are assumed to be known and independent of the price charged for the product. Third, there is unlimited capacity to ship across any particular transportation route. Fourth, the problem deals with a single commodity or an unchanging mix of multiple commodities.

These assumptions have spawned many extensions, including for example, the transshipment problem (Orden), wherein transshipment through intermediate cities is permitted. Another extension allows the quantity supplied and demanded to depend on price. This problem is called a spatial equilibrium model (Takayama and Judge(1973)) and is covered in the price endogenous chapter. Problems also have been formulated with capacitated transportation routes where simple upper bounds are placed on the shipment from a supply point to a demand point (i.e.,  $X_{ij}$  less than or equal to  $UL_{ij}$ ). These problems are generally in the purview of network theory (Bazarrá et. al., Kennington). Multi-commodity transportation problems have also been formulated (Kennington). Cost/volume relationships have been included as in the warehouse location model in the second integer programming chapter. Finally, the objective function may be defined as containing more than just transportation costs. Ordinarily one thinks of the problem wherein the  $c_{ij}$  is the cost of transporting goods from supply point  $i$  to demand point  $j$ . However, modelers such as Barnett et al. (1984) have included the supply cost, so the overall objective function then involves minimizing delivered cost. Also the transport cost may be defined as the demand price minus the transport cost minus the supply price, thereby converting the



problem into a profit maximization problem.

The transport model has interesting solution properties. The constraint matrix possesses the mathematical property of unimodularity (see Bazaraa et al. (1984) for explanation) which causes the solution to be integer-valued when the right hand sides are integers. Also, when the sum of the supplies equals the sum of the demands, then the problem is degenerate, and the dual will have multiple optimal solutions. That is, the supply and demand prices will not be unique.

The transportation problem has been the subject of considerable research and application. The research has led to special solution approaches including the Ford and Fulkerson out of kilter algorithm, and specializations of the primal simplex algorithm. Discussion of such topics can be found in Dantzig (1951), Bradley et al., and Glover et al. (1974). It is worthwhile to point out that the algorithms allow the problem to be solved at a factor of 100 times faster than the use of the general simplex method. Readers attempting to solve transportation problems should consult the network literature for efficient solution procedures.

There have been many applications of different versions of the transportation problem. For example, it has been used to study the effect of railroad regulatory policy (Baumel et al.), grain subterminal facility location (Hilger et al.), and port location (Barnett et al.(1984)). The assignment and/or contract award problems are transport problems which arose early in the development of LP (see the assignment and contract awards sections in Riley and Gass). There also are related formulations such as the caterer problem.

#### **5.4 Diet/Feed Mix/Blending Problem**

One of the earliest LP formulations was the diet problem along with the associated feed mix and blending variants. The diet context involves composing a minimum cost diet from a set of available ingredients while maintaining nutritional characteristics within certain bounds. Usually, a total dietary volume constraint is also present. Stigler studied the problem before LP was developed. However, he noted that tastes

and preferences cause a disparity between observed and minimum cost human diets. Basically the human diet form of this problem takes a largely expository role with few applications. However, Waugh applied LP to the livestock feed formulation problem, and it has become one of the most widely used LP applications. The formulation has also been applied to the blending of ice cream, sausage, and petroleum (Riley and Gass).

The basic model is formulated as follows. Define index (  $i$  ) representing the nutritional characteristics which must fall within certain limits (protein, calories, etc.). Define index (  $j$  ) which represents the types of feedstuffs available from which the diet can be composed. Then define a variable (  $F_j$  ) which represents how much of each feedstuff is used in the diet. The constraints of the problem include the normal nonnegativity restrictions plus three additional constraint types: one for the minimum requirements by nutrient, one for the maximum requirements by nutrient and one for the total volume of the diet. In setting up the nutrient based constraints parameters are needed which tell how much of each nutrient is present in each feedstuff as well as the dietary minimum and maximum requirements for that nutrient. Thus, let  $a_{ij}$  be the amount of the  $i^{\text{th}}$  nutrient present in one unit of the  $j^{\text{th}}$  feed ingredient; and let  $UL_i$  and  $LL_i$  be the maximum and minimum amount of the  $i^{\text{th}}$  nutrient in the diet. Then the nutrient constraints are formed by summing the nutrients generated from each feedstuff ( $a_{ij}F_j$ ) and requiring these to exceed the dietary minimum and/or be less than the maximum. The resultant constraints are

$$\sum_j a_{ij}F_j \leq UL_i$$

$$\sum_j a_{ij}F_j \geq LL_i$$

A constraint is also needed that requires the ingredients in the diet equal the required weight of the diet. Assuming that the weight of the formulated diet and the feedstuffs are the same, this requirement can be written as

$$\sum_j F_j = 1.$$

Finally an objective function must be defined. This involves definition of a parameter for feedstuff cost ( $c_j$ ) and an equation which sums the total diet cost across all the feedstuffs, i.e.,

$$\sum_j c_j F_j$$

The resulting LP formulation is

$$\begin{aligned} \text{Min} \quad & \sum_j c_j F_j \\ \text{s.t.} \quad & \sum_j a_{ij} F_j \leq UL_i \quad \text{for all } i \\ & \sum_j a_{ij} F_j \geq LL_i \quad \text{for all } i \\ & \sum_j F_j = 1 \\ & F_j \geq 0 \quad \text{for all } j \end{aligned}$$

This formulation depicts a cost minimization problem. The  $F_j$  activities provide an example of purchase variables, depicting the purchase of one unit of feed ingredients at an exogenously specified price which, in turn, provides the nutrient characteristics in the mixed diet. The constraints are in the form of resource limits and minimum requirements.

The dual of this problem contains variables giving the marginal value of the nutrient upper and lower limits, as well as the value of the overall volume constraint. The dual is

$$\begin{aligned}
\text{Max} \quad & \sum_i (\lambda_i U_{L_i} - \mu_i L_{L_i} - \nu) \\
\text{s.t.} \quad & \sum_i (\lambda_i a_{ij} - \mu_i a_{ij} - \nu) \leq c_j \text{ for all } j \\
& \lambda_i, \mu_i \geq 0 \text{ for all } i \\
& \nu \text{ unrestricted}
\end{aligned}$$

The dual variables are  $\lambda_i$ , the marginal value of the  $i^{\text{th}}$  nutrient upper limit constraint;  $\mu_i$ , the marginal value of the  $i^{\text{th}}$  nutrient lower limit constraint; and  $\nu$ , the marginal value of the total volume constraint. The dual constraints require that the imputed cost of each ingredient plus the cost of the volume constraint to be no greater than the cost of any ingredient. The variables  $\lambda_i$  and  $\mu_i$  are nonnegative, but  $\nu$  is unrestricted in sign. Since the primal constraints impose nutrient requirements, the dual objective function demonstrates how the costs of the diet are allocated to individual nutritional requirements. For further economic interpretation of the dual objective function, see Thomas et al.

#### 5.4.1 Example

Suppose we use the model in a setting where we value a potential new ingredient in the diet. Suppose that cattle feeding involves lower and upper limits on net energy, digestible protein, fat, vitamin A, calcium, salt and phosphorus. Further, suppose the feed ingredients available are corn, hay, soybeans, urea, dical phosphate, salt and vitamin A. In addition, a new product, potato slurry, is available. One kilogram of the feed is to be mixed. The costs of the ingredients per kilogram (excluding slurry) are shown in Table 5.11. The nutrient requirements are given in Table 5.12.

The nutrient requirements give the minimum and maximum amounts of each nutrient in one kilogram of feed. Thus, there must be between 0.071 and 0.130 kg of digestible protein in one kg of feed. The volume of feed mixed must equal one kilogram. The nutrient compositions of one kg of each potential feed are shown in Table 5.13. The formulation is shown in Table 5.14. The reader should also note that the potato slurry

activity has been inserted with a price of P. This price will be varied.

First, suppose we specify the price of potato slurry to be one cent per kilogram, then the GAMS formulation is given in Table 5.15 and the file DIET. Three sets are involved, the first two are equivalent to the subscripts for ingredients and nutrients. The third one is one for convenience in data input and simply tells whether a minimum or maximum is being defined. In turn, the cost, minimum, maximum, and the nutrient content of each of the feedstuffs are defined. The variables identify the use of each feedstuff and a total cost variable for the objective function. The nutrient upper and lower limits, nutrients and the weight constraint are defined in lines 39-45. Then the model solve statements appear. In addition, there are instructions which appear between the ontext and offtext statements that run the model repeatedly through a LOOP statement to generate a series of related solutions for increasing potato slurry prices. This section also contains an example of report writing. Namely, 30 slurry price scenarios are defined, and then a set is defined where the results of the 30 price runs are kept. The solution printout is suppressed (by the OPTION statement) and only the summary result is reported. Inside the LOOP a calculation sets the ingredient cost equal to the previous price plus a half cent for each scenario. In turn, the model is solved for each price level, with slurry use and shadow price recorded in the table. After the loop is finished, the table with all the results is displayed. Note the report writing features in line 55 where the .L parameter on the end of the feed use variable indicates that the optimal value of the FEED variable for slurry is recorded in a table.

The solution at a slurry price of 0.01 yields an objective value of 0.021 with the variable and constraint solutions as shown in Table 5.16. The optimal solution indicates that the least cost feed ration is 95.6 percent slurry, 0.1 percent vitamin A, 1.5 percent salt, 0.2 percent dicalcium phosphate, 1.4 percent urea, 1.1 percent soybeans and 0.1 percent hay. The shadow prices indicate that the binding constraints are the net energy, vitamin A, salt, and calcium minimums along with the phosphorous maximum and the weight constraint. The shadow price on the weight constraint is 1.08 cents.

The results using the code in lines 49-57 show that the above solution would hold for prices of potato slurry between 0 and +11 cents. A second ration with 87% slurry would hold for prices between roughly 11 and 12 cents. A ration with 18% slurry would hold for prices around 12.5 cents, and for any price greater than 13 cents no potato slurry would be used. The resultant demand schedule for potato slurry is graphed in Figure 5.1.

The empirical dual formulation is given in Table 5.17 and the file DUALDIET. This formulation, with  $P$  equal to 0.01, leads to a dual solution with the same objective function value as in the primal. The optimal dual solution is shown in Table 5.18. The dual is degenerate.

#### **5.4.2 Comments**

There are three assumptions within the feed formulation problem. First, the nutrient requirements are assumed constant and independent of the final product (e.g., livestock) price. Second, the quality of each feed ingredient is known. Third, the diet is assumed to depend on only feed price and nutrients.

The diet problem is widely used, especially in formulating feed rations. Animal scientists use the term "ration-balancing", and several software programs have been specifically developed to determine least cost rations. LP models of the ration problem were used by Prevatt, et al., to evaluate fed cattle production in Florida and by Thomas et al. (1992) who examined nutrient values in dairy production. Other references and discussions are given in Dorfman et al., Ashour and Anderson, and Beneke and Winterboer.

### **5.5 Joint Products**

Many applied LP models involve production of joint products. An example would be a petroleum cracking operation where production yields multiple products such as oil and naphtha. Other examples include dairy production where production yields both milk and calves, or forestry processing where trees yield sawdust and multiple types of sawn lumber. Here, we present a formulation explicitly dealing with joint products.

Suppose we articulate the problem as follows. Consider a profit maximizing firm who wishes to maximize total profits derived from the available production possibilities. Furthermore, suppose each of the production possibilities yields multiple products, uses some inputs with a fixed market price, and uses some resources that are available in fixed quantity.

Formulation of this model requires indices which depict the products which could be produced ( $p$ ), the production possibilities ( $j$ ), the fixed price inputs purchased ( $k$ ), and the resources which are available in fixed quantity ( $m$ ). Three types of variables need to be defined. The first ( $X_p$ ) defines the total quantity of the  $p^{\text{th}}$  product sold; the second ( $Y_j$ ) identifies the quantity of  $j^{\text{th}}$  production possibility utilized; and the third ( $Z_k$ ) is the amount of the  $k^{\text{th}}$  input purchased.

Other than nonnegativity, three types of constraints are needed. The first relates the quantity sold of each product to the quantity yielded by production. Algebraic specification requires definition of a parameter ( $q_{pj}$ ) which gives the yield of each product ( $p$ ) by each production possibility. The expression

$$X_p - \sum_j q_{pj} Y_j \leq 0$$

is a supply demand balance. Here demand, in the form of sales of  $p^{\text{th}}$  product, is required to be less than or equal to supply which is the amount generated across the production alternatives. Further, since production and sales are endogenous, this is written as sales minus production and is less than or equal to zero.

The second type of constraint relates the quantity purchased of each fixed price input to the quantity utilized by the production activities. The parameter  $r_{kj}$  gives the use of the  $k^{\text{th}}$  input by the  $j^{\text{th}}$  production possibility. In turn, the constraint sums up total fixed price input usage and equates it to purchases as follows:

$$\sum_j r_{kj} Y_j - Z_k \leq 0.$$

This constraint is another example of a supply demand balance where the endogenous demand in this case, use of the  $k^{\text{th}}$  input, is required to be less than or equal to the endogenous supply which is the amount purchased.

The third type of constraint is a classical resource availability constraint which insures that the quantity used of each fixed quantity input does not exceed the resource endowment. Specification requires definition of parameters for the resource endowment ( $b_m$ ) and resource use when the production possibility is utilized. The constraint which restricts total resource usage across all possibilities is

$$\sum_j s_{mj} Y_j \leq b_m$$

where  $s_{mj}$  is the use of the  $m$ th resource by  $Y_j$ .

For the objective function, an expression is needed for total profits. To algebraically expressed the profits require parameters for the sales price ( $c_p$ ), the input purchase cost ( $e_k$ ), and any other production costs associated with production ( $d_j$ ). Then the objective function can be written as

$$\sum_p c_p X_p \quad \& \quad \sum_j d_j Y_j \quad \& \quad \sum_k e_k Z_k$$

The individual terms do not reflect the profit contribution of each variable in an accounting sense, rather this occurs across the total model. Thus, the production variable term ( $d_j$ ) does not include either the price of the products sold or the cost of all the inputs purchased, but these components are included by terms on the sales and purchase variables. The resultant composite joint products model is

$$\begin{aligned} \text{Max} \quad & \sum_p c_p X_p \quad \& \quad \sum_j d_j Y_j \quad \& \quad \sum_k e_k Z_k \\ \text{s.t.} \quad & X_p \quad \& \quad \sum_j q_{pj} Y_j \quad \# \quad 0 \quad \text{for all } p \\ & \sum_j r_{kj} Y_j \quad \& \quad Z_k \quad \# \quad 0 \quad \text{for all } k \\ & \sum_j s_{mj} Y_j \quad \# \quad b_m \quad \text{for all } m \\ & X_p, \quad Y_j, \quad Z_k \quad \geq \quad 0 \quad \text{for all } p, j, k \end{aligned}$$

Several features of this formulation are worth mention. First, note the explicit joint product relationships. When activity  $Y_j$  is produced, a mix of joint outputs ( $q_{pj}$ ,  $p=1, 2, \dots$ ) is produced while



simultaneously consuming the variable inputs both directly priced in the objective function ( $d_j$ ) and explicitly included in constraints ( $r_{kj}$ ), along with the fixed inputs ( $s_{mj}$ ). Thus, we have a multi-factor, multi-product production relationship.

Another feature of this problem involves the types of variables and constraints which are used. The variables  $X_p$  are sales variables which sell the available quantities of the outputs. The variables  $Z_k$  are purchase variables which supply the inputs utilized in the production process. The variables  $Y_j$  are production variables. In this case, the production variables show production explicitly in the matrix, and the product is sold through another activity. The first two constraints are supply-demand balances. The reason for the use of inequalities in supply/demand balances is discussed in Chapter VII. The last constraint is a resource endowment.

The dual to this LP problem is

$$\begin{array}{ll}
 \text{Min} & \sum_m U_m b_m \\
 \text{s.t.} & V_p \leq c_p \quad \text{for all } p \\
 & \sum_p V_p a_{pj} - \sum_k W_k r_{kj} - \sum_m U_m s_{mj} \leq d_j \quad \text{for all } j \\
 & W_k \leq e_k \quad \text{for all } k \\
 & V_p, W_k, U_m \geq 0 \quad \text{for all } p, k, m
 \end{array}$$

where  $V_p$  is the marginal value of product  $p$ ;  $W_k$  the marginal cost of the  $k^{\text{th}}$  variable input; and  $U_m$  is the marginal value of the  $m^{\text{th}}$  fixed resource.

The dual objective function minimizes total marginal value of the fixed resources. The first constraint insures that the marginal value of each product is at least as great as the sales price. This relationship demonstrates the dual implication of a sales variable; namely, a lower bound is imposed on the shadow price for the commodity sold. The third constraint insures that the marginal value of each input is no more than its purchase price and reflects the dual implication of a purchase variable, i.e., an upper bound is imposed on the shadow price of the item which can be purchased. The second constraint is allied with the production variable and insures that the total value of the products yielded by a process is less than or equal to the value of the inputs used in production, where the input valuation involves the direct costs in the objective function plus the imputed costs due to variable and fixed inputs.

### 5.5.1 Example

Consider a farm which produces both wheat and wheat straw using seven production processes. The basic data for these production processes are given in Table 5.19. The production process involves the joint production of wheat and straw using land, seed and fertilizer.

The relevant prices are wheat - \$4.00 per bushel, wheat straw - \$.50 per bale, seed - \$.20/lb., and fertilizer - \$2.00 per kilogram. Also there is a \$5 per acre production cost for each of the processes and the farm has 500 acres. This leads to the empirical formulation

$$\begin{array}{l}
 \text{Max} \quad 4X_1 + .5X_2 - 5Y_1 - 5Y_2 - 5Y_3 - 5Y_4 - 5Y_5 - 5Y_6 - 5Y_7 - 2Z_1 - .2Z_2 \\
 \text{s.t.} \quad X_1 \quad \quad \quad - 30Y_1 - 50Y_2 - 65Y_3 - 75Y_4 - 80Y_5 - 80Y_6 - 75Y_7 \quad \quad \quad \# \quad 0 \\
 \quad \quad \quad X_2 - 10Y_1 - 17Y_2 - 22Y_3 - 26Y_4 - 29Y_5 - 31Y_6 - 32Y_7 \quad \quad \quad \# \quad 0 \\
 \quad \quad \quad \quad \quad \quad + 5Y_2 + 10Y_3 + 15Y_4 + 20Y_5 + 25Y_6 + 30Y_7 - Z_1 \quad \quad \quad \# \quad 0 \\
 \quad \quad \quad \quad \quad \quad 10Y_1 + 10Y_2 + 10Y_3 + 10Y_4 + 10Y_5 + 10Y_6 + 10Y_7 \quad \quad \quad - Z_2 \# \quad 0 \\
 \quad \quad \quad \quad \quad \quad Y_1 + Y_2 + Y_3 + Y_4 + Y_5 + Y_6 + Y_7 \quad \quad \quad \# \quad 500 \\
 X_1, X_2, Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, Y_7, Z_1, Z_2 \geq 0
 \end{array}$$

The variables  $X_1$  and  $X_2$  are the sales of wheat and straw, respectively. The variables  $Y_1$  through  $Y_7$  are the production of wheat via the seven different input usage/output production possibilities. The variables

$Z_1$  and  $Z_2$  are purchases of fertilizer and seed, respectively. The first two constraints balance product sale with production; the third and fourth constraints require purchases of seed and fertilizer when production is undertaken. The fifth constraint limits land availability. Note that the  $Y$  variables produce the multiple products using multiple inputs. This problem exhibits seven different ways of producing the two products. Each of these seven different ways uses a different input combination. Also note that by increasing the fertilizer input one obtains decreasing returns in terms of the production of wheat and straw.

The GAMS formulation for this problem is given in Table 5.20 and the file JOINT. This formulation relies on four sets. One for the products; one for the purchased inputs; one for the fixed inputs; and one for the possible production processes. In turn, parameters are specified for the product prices, input costs, the production process costs, and fixed resource availability. These data are entered conventionally except for the constant production cost which is entered algebraically in line 11.

Variables and constraints are defined as associated with the above algebraic formulation, with the net income variable specified to equal the objective function. Similarly, the objective function, yield balance, input balance and available resource constraints are defined. Then the model is defined indicating all constraints and solved.

The solution to this problem is shown in Table 5.21. The solution implies that 40,000 bushels of wheat and 14,500 bales of straw are produced by 500 acres of the fifth production possibility using 10,000 kilograms of fertilizer and 5,000 lbs. of seed. The reduced cost information shows a \$169.50 cost for the first production possibility if undertaken. Under this production pattern, the marginal value of land is \$287.50. The shadow prices on the first four rows are the sale and purchase prices of the various outputs and inputs depicted in those rows.

The dual of the joint product example is given in Table 5.22. Note the action of the constraints. The first two insure that the shadow prices associated with the balance row for each product are at least as great as

the product prices. This is a clear condition for optimality. If the shadow price fell below this, it would be rational to sell more of the product. The last two constraints insure that the values of the inputs are less than or equal to their purchase prices. If the input was worth more than its purchase price, it would make sense to purchase more until the value fell to the purchase price. The middle constraints, under re-arrangement, insure that the total imputed value of all the products from any one production method is less than or equal to the value of the fertilizer, seed and land used arising through the terms involving  $W_1$ ,  $W_2$  and  $U$ . The optimal solution to the dual problem is given in Table 5.23 and corresponds exactly to the optimal primal solution.

### 5.5.2 Comments

The joint products problem illustrates: 1) the proper handling of joint products and 2) production variables where the returns from production are not collapsed into the objective function but explicitly appear in the constraints.

The formulation also illustrates the possible complexity of LP. In this case product balance constraints are incorporated in a model along with resource constraints. Also note that  $X_1$  and  $X_2$ , gives the sum of total output, and that  $Z_1$  and  $Z_2$  give the sum of total input usage on the farm which may be convenient for model interpretation. It is possible to exclude the  $X$  and  $Z$  variables by adjusting the objective function coefficients on  $Y$  so they reflect the value of the products ( $Y$ ) less the costs of the inputs ( $Z$ ). The larger formulation used above may be superior if model use demands easy alteration of prices or summary accounting on total production. This is discussed in the Purposeful Modeling section of the Toward Proper Modeling chapter.

Another observation is the action of the primal variables as reflected in the dual. Note that the purchase activities provide upper bounds for the dual variables associated with the output balance constraint, while the sales activities provide lower bounds for the dual variables associated with the input balance constraints, and the production activities provide relationships between the dual variables relating to inputs and

outputs.

Joint product formulations have a relatively long history. It is difficult to cite many exact applications; rather such a structure is common and implicit in many models throughout the literature. Hildreth and Reiter presented an early joint products example, Beneke and Winterboer provide examples of joint products, (e.g. p. 46), and many other examples are available.

## **5.6 Comments on GAMS and Algebraic Modeling**

It is beneficial to examine the advantages and disadvantages of this book's orientation toward algebraic modeling and GAMS usage. This is done in this section.

### **5.6.1 Why Use Algebraic Modeling?**

Algebraic modeling refers to the formulation of optimization problems in summation notation. Why is algebraic modeling desirable and what are its inherent shortcomings? The major advantages involve the ability to concisely state problems in an abstract general fashion, largely independent of the data and exact application context while making general duality implication statements. One can produce an algebraic formulation independent of the problem size which initially can be used to consider moving products from two plants to two regions and later to consider moving products from fifty plants to a hundred regions. Such an approach is particularly useful in the textbook presentation herein. Furthermore, familiarity with algebraic modeling allows the use of formulations from the literature.

Nevertheless, these advantages are accompanied by several disadvantages. Algebraic modeling and summation notation are difficult for some users. Some people will always desire to deal with the exact problem context, not an abstract general formulation. This does lead to a strategy most modelers use when employing algebraic modeling. Namely, algebraic modeling exercises are usually supported by small tableau based sample models which capture problem essence and serve as an aid in algebraic model formulation. Dual

model setup and interpretation can also be difficult. Empirically based duals are usually easier to construct than algebraic duals.

Algebraic modeling can be quite cumbersome when dealing with small, one time problems. In such cases, the generality of the presentation may not be worth the effort. Clearly, the gains to using algebraic modeling rise with problem size. When a modeler deals with large problems, the algebraic statement is probably the only thing that is thoroughly understood. Often the numerical formulation has grown out of control. Another problem when using algebraic modeling is that certain tableau based formulation structures are difficult to express algebraically. This is illustrated by the sequencing example of Chapter 7.

Algebraic modeling is not the tool of choice for small, infrequently solved problems, but rather it is best employed for medium or large sized models (more than 30 rows and/or columns) although the GAMS report writing and computation features may make it attractive for small models. In such cases, algebraic modeling can allow the modeler to gain general insights into the problem. Furthermore, coupling algebraic modeling with a modeling system like GAMS permits gains in modeling efficiency and efficacy.

### **5.6.2 Why Use GAMS to Support Modeling?**

Now, why use GAMS? GAMS generally requires algebraic modeling and thus, has many of algebraic modeling's advantages and disadvantages. However, GAMS is easier to tailor to the problem, so its use introduces additional advantages and overcomes some of the disadvantages.

The crucial point when considering whether to use GAMS involves a comparison with traditional linear programming formulation techniques. Here we compare GAMS usage with: a) direct specification of equation-based tableaus as used in LINDO (Schrage); b) direct specification of full tableaus as in McMillan; and c) MPS-based input as in the stand alone version of MINOS (Murtaugh and Saunders).

First, let us deal with data input. Given, the resource allocation problem, alternative formulations for GAMS, a tableau-based approach as in McMillan, a LINDO input stream (Schrage (1985)), a MPS input

stream (Murtaugh and Saunders(1987)) and an alternative simplified GAMS input stream appear in Table 5.24.

Several features appear in these input streams. First, the algebraic GAMS is the most complex. Second, GAMS can be used in a much simpler fashion becoming competitive with LINDO. Third, GAMS is more readable than the tableau or MPS. Fourth, the versatility of GAMS is shown by the fact that the structure of the resource allocation problem from the earlier example is maintained with a different data set (in Panel a).

Now let us deal with GAMS disadvantages. Specification of a small model almost always takes a larger file in GAMS than when using traditional equation or tableau-based methods. Namely, the LINDO or McMillan code specification of the resource allocation example takes three to five lines of input, whereas the GAMS specification takes the better part of a page. However, GAMS can be used simplistically yielding almost the same as LINDO (Panel d). Furthermore, a higher level of computer skill is required with GAMS as a batch input file, a compiler and a 250 page users guide are involved. However GAMS is generally easier to use than MPS-based methods, as an inspection of the treatment in Beneke and Winterboer or Orchard-Hays reveals.

There are also model formulation characteristics which constitute disadvantages. GAMS generally requires everything to be written algebraically which makes some potential users uncomfortable. Also some problem structures are difficult to write algebraically (i.e., see the sequencing problem in chapter VII).

Finally, there are shortcomings due to a lack of capabilities in GAMS. GAMS does not contain a set of well-developed model debugging aids as are present in many MPS-based installations. One can add custom diagnostic aids only with difficulty (The GAMS IO Library document explains how to interface FORTRAN, PASCAL and C programs through a special programming job). It is also difficult to construct an extremely problem-tailored computer implementation as can be done using FORTRAN-based computer implementation

(i.e., see McCarl et al.(1977) or McCarl and Nuthall).

### Aid with Initial Problem Formulation and Changes in Problem Formulation

GAMS aids both in initially formulating and subsequently revising formulations. GAMS facilitates specification and debugging of an initial formulation by allowing the modeler to begin with a small data set, then after verifying correctness expand to a much broader context. For example, one could initially specify a small transportation model with a few suppliers and demanders. Then after that model is debugged one could respecify the problem to encompass fifty shipping origins and two hundred destinations without needing to change the algebraic model.

Second, GAMS makes it easy to alter the model. Historically large models have been difficult to modify. Namely when using traditional MPS-based modeling approaches it is difficult to add new constraints or variables. On the other hand, GAMS allows one to add model features much more simply. Generally, modelers do not try to make a complete formulation the first time around, rather one starts with a small formulation and then adds structural features as demanded by the analyst. GAMS also enforces consistent modeling, allowing models to be transferred between problem contexts (i.e., code from the transport example is used in the warehouse location example in the second integer programming chapter).

### Automated Computational Tasks

Many of the tasks that would traditionally have required a computer programmer are automated. As such, GAMS automatically does coefficient calculation; checks the formulation for obvious flaws; chooses the solver; formats the programming problem to meet the exact requirements of the solver; causes the solver to execute the job; saves and submits the advanced basis when doing related solutions; and permits usage of the solution for report writing. Also GAMS verifies the correctness of the algebraic model statements and allows empirical verification. Furthermore, since GAMS has been implemented on machines ranging from PCs to workstations to CRAY super computers, it allows portability of a model formulation between computer



systems. Switching solvers is also very simple requiring changing a solver option statement or changing from using LP to using NLP. Links to spreadsheets have also been developed. Finally, GAMS does not require programming using languages like C or FORTRAN as does traditional MPS-based matrix generators and report writers.

#### Facilitates Documentation and Later Use of Models

One of the other convenient features of GAMS is its self-documenting nature. Perhaps the largest benefit GAMS has over traditional modeling techniques is that modelers can use longer variable, equation and index names as well as comments, data definitions etc., allowing a more adequate and readable problem description. Model structure, assumptions, and any calculation procedures used in the report writing are documented as a byproduct of the modeling exercise in a self-contained file. Comment statements can be inserted by placing an asterisk in column one, followed by text identifying data sources or particular assumptions being used (i.e., in some of the authors' models, comments identify data source publication and page). Under such circumstances GAMS allows either the original author or others to alter the model structure and update data.

#### Allows Use by Varied Personnel

Modeling personnel are often rare. For example, in international development contexts, detailed GAMS applications have been set-up by modeling experts but subsequently, the model is utilized by policy-makers with minimal, if any, assistance from the modeling experts. Often, given proper internal documentation and a few instructions, clerical labor and nontechnical problem analysts can handle an analysis.

#### Supported by Libraries of Applications

GAMS is rapidly becoming a defacto industry standard. Many examples are provided by the model library, Thompson and Thore; Kendrick; Zenios and their material.

## References

- Ashour, S. and C. Anderson. "Linear Programming Analysis of a Cattle Feedlot." In H.M. Salkin and J. Saha (eds.). Studies in Linear Programming. New York: North Holland Co., 1975.
- Barnett, D., J. Binkley, and B. McCarl. "Port Elevator Capacity and National and World Grain Shipments." Western Journal of Agricultural Economics. 9(1984):77-89.
- Baumel, C.P., J.J. Miller, and T.P. Drinka. "The Economics of Upgrading Seventy-one Branch Rail Lines in Iowa." American Journal of Agricultural Economics. 59(1977):61-70.
- Beneke, R.R. and R. Winterboer. Linear Programming Applications to Agriculture. Ames: The Iowa State University Press, 1973.
- Bradley, S.P., A.C. Hax and T.L. Magnanti. Applied Mathematical Programming. Reading, MA: Addison-Wesley Publishing Company, Inc., 1977.
- Brooke, A., D. Kendrick and A. Meeraus. GAMS: A User's Guide. Boyd and Fraser Publishers, Version 2.25.
- Dantzig, G.B. "Application of the Simplex Method to a Transportation Problem." In T.C. Koopmans (ed.), Activity Analysis of Production and Allocation. New York: John Wiley and Sons, Inc., 1951, pp. 359-373.
- Dorfman, R. Application of Linear Programming to the Theory of the Firm. Berkeley: University of California Press, 1951.
- Dorfman, R., P.A. Samuelson, and R.M. Solow. Linear Programming and Economic Analysis. New York: McGraw-Hill Book Company, Inc., 1958.
- Ford, L.R., Jr. and D.R. Fulkerson. Flows in Networks. Princeton, NJ: Princeton University Press, 1962.
- Glover, F., D. Karney, and D. Klingman. "Implementation and Computational Comparisons of Primal, Dual and Primal-Dual Computer Codes for Minimum Cost Network Flow Problems." Networks. 4(1974):191-212.
- Hildreth, C.G. and S. Reiter. "On the Choice of a Crop Rotation Plan." in T.J. Koopmans (ed.) Activity Analysis of Production and Allocation. Cowles Commission Monograph 13. New York: John Wiley and Sons, 1951.
- Hilger, D., B. McCarl, and J. Uhrig. "Facilities Location: The Case of Grain Subterminals." American Journal of Agricultural Economics. 59:(1977):674-681.
- Hitchcock, F.L. "The Distribution of a Product from Several Sources to Numerous Localities." Journal of Mathematical Physics. 20(1941):244-230.

- Kantorovich, L.V. "Mathematical Methods in the Organization and Planning of Production." Publication House of the Leningrad State University, 1939. 68 pp. Translated in Management Science. 6(1960):366-422.
- Kendrick, D.A. Models for Analyzing Comparative Advantage. Norwell, MA: Kluwer Academic Publishers, 1990.
- Kennington, J.L. "A Survey of Linear Cost Multicommodity Network Flows." Operations Research. 26(1978):209-236.
- Koopmans, T.C. "Optimum Utilization of the Transportation System." Proceedings of the International Statistical Conferences. (Vol. 5 reprinted as Supplement to Econometrica Vol. 17, 1949). Washington, DC, 1947.
- McCarl, B.A., W.V. Candler, D.H. Doster, and P.R. Robbins. "Experiences with Farmer Oriented Linear Programming for Crop Planning." Canadian Journal of Agricultural Economics. 25(1977):17-30.
- McCarl, B.A. and P. Nuthall. "Linear Programming for Repeated Use in the Analysis of Agricultural Systems." Agricultural Systems. 8,1(1982):17-39.
- McMillan, C. Mathematical Programming: An Introduction to the Design and Application of Optimal Decision Machines. New York: John Wiley and Sons, Inc., 1970.
- Murtaugh, B.A. and M.A. Saunders. "MINOS 5.1 User's Guide." Report SOL 83-20R, December 1983, revised January 1987, Stanford University.
- Orchard-Hays, W. Advanced Linear Programming Computing Techniques. New York: McGraw-Hill, 1968.
- Orden, A. "The Transshipment Problem." Management Science. 2(1956):276-85.
- Prevatt, J.W., B.E. Melton, T.H. Spreen, W.K. Mathis. "The Economics of Carcass Beef Production: and Appraisal of Florida's Feedlot Potential." Southern Journal of Agricultural Economics. 10,2(1978):49-55.
- Riley, V. and S.I. Gass. Linear Programming and Associated Techniques. Published for Operations Research Office. Baltimore, MD: The Johns Hopkins Press, 1958.
- Salkin, H.M. and J. Saha. Studies in Linear Programming. New York: American Elsevier, 1975.
- Schrage, L. Linear, Integer, and Quadratic Programming with LINDO: User's Manual. Second Edition. The Scientific Press, 1985.
- Stigler, G.J. "The Cost of Subsistence." Journal of Farm Economics. 27(1945):303-314.
- Takayama, T. and G. Judge. Studies in Economic Planning over Space and Time. New York: North Holland

Co., 1973.

Thomas, C., M.A. DeLorenzo, D.K. Beede, and T.H. Spreen. "Predicting Daily Feed Costs for Dairy Management Models." Journal of Dairy Science. Forthcoming, 1992.

Thompson, Gerald L. and Sten Thore. Computational Economics: Economic Modeling with Optimization Software. San Francisco, CA: The Scientific Press, 1992.

Waugh, F.V. "The Minimum-Cost Dairy Feed." Journal of Farm Economics. 33(1951):299-310.

Williams, H.P. Model Building in Mathematical Programming. 2nd Edition. Chichester; New York: John Wiley & Sons, 1985.

Zenios, S.A., editor, 1992, "Financial Optimization," Cambridge University Press.

**Table 5.1. Resource Requirements for the E-Z Chair Makers for the Normal Pattern**

	Hours of Use per Chair Type	
	Functional	Fancy
Small Lathe	0.8	1.2
Large Lathe	0.5	0.7
Chair Bottom Carver	0.4	1.0
Labor	1.0	0.8

**Table 5.2. Resource Requirements and Increased Costs for Alternative Methods to Produce Functional and Fancy Chairs in Hours of Use per Chair and Dollars**

	Maximum Use of Small Lathe		Maximum Use of Large Lathe	
	Functional	Fancy	Functional	Fancy
Small Lathe	1.30	1.70	0.20	0.50
Large Lathe	0.20	0.30	1.30	1.50
Chair Bottom Carver	0.40	1.00	0.40	1.00
Labor	1.05	0.82	1.10	0.84
Cost Increase	\$1.00	\$1.50	\$0.70	\$1.60

**Table 5.3. First GAMS Formulation of Resource Allocation Example**

```

1  SET  PROCESS      TYPES OF PRODUCTION PROCESSES
2                                     /FUNCTNORM , FUNCTMXSML , FUNCTMXLRG
3                                     ,FANCYNORM , FANCYMXSML , FANCYMXLRG/
4      RESOURCE      TYPES OF RESOURCES
5                                     /SMLLATHE,LRGLATHE,CARVER,LABOR/ ;
6
7  PARAMETER PRICE(PROCESS)  PRODUCT PRICES BY PROCESS
8                                     /FUNCTNORM 82, FUNCTMXSML 82, FUNCTMXLRG 82
9                                     ,FANCYNORM 105, FANCYMXSML 105, FANCYMXLRG 105/
10     PRODCOST(PROCESS)  COST BY PROCESS
11                                     /FUNCTNORM 15, FUNCTMXSML 16 , FUNCTMXLRG 15.7
12                                     ,FANCYNORM 25, FANCYMXSML 26.5, FANCYMXLRG 26.6/
13     RESORAVAIL(RESOURCE) RESOURCE AVAILABILITY
14                                     /SMLLATHE 140, LRGLATHE 90,
15                                     CARVER 120, LABOR 125/;
16
17  TABLE RESOURUSE(RESOURCE,PROCESS) RESOURCE USAGE
18
19     FUNCTNORM  FUNCTMXSML  FUNCTMXLRG
20  SMLLATHE      0.80      1.30      0.20
21  LRGLATHE      0.50      0.20      1.30
22  CARVER         0.40      0.40      0.40
23  LABOR          1.00      1.05      1.10
24  +             FANCYNORM  FANCYMXSML  FANCYMXLRG
25  SMLLATHE      1.20      1.70      0.50
26  LRGLATHE      0.70      0.30      1.50
27  CARVER         1.00      1.00      1.00
28  LABOR          0.80      0.82      0.84;
29
30  POSITIVE VARIABLES
31     PRODUCTION(PROCESS) ITEMS PRODUCED BY PROCESS;
32  VARIABLES
33     PROFIT          TOTALPROFIT;
34  EQUATIONS
35     OBJT            OBJECTIVE FUNCTION ( PROFIT )
36     AVAILABLE(RESOURCE)  RESOURCES AVAILABLE ;
37
38  OBJT..  PROFIT =E=
39     SUM(PROCESS,(PRICE(PROCESS)-PRODCOST(PROCESS))
40         * PRODUCTION(PROCESS)) ;
41
42  AVAILABLE(RESOURCE)..
43     SUM(PROCESS,RESOURUSE(RESOURCE,PROCESS)*PRODUCTION(PROCESS))
44         =L= RESORAVAIL(RESOURCE);
45
46  MODEL RESALLOC /ALL/;
47  SOLVE RESALLOC USING LP MAXIMIZING PROFIT;

```

**Table 5.4. Second GAMS Formulation of Resource Allocation Example**

```

1 SET CHAIRS TYPES OF CHAIRS /FUNCTIONAL,FANCY/
2 PROCESS TYPES OF PRODUCTION PROCESSES
3 /NORMAL , MXSMLLATHE , MXLRGLATHE/
4 RESOURCE TYPES OF RESOURCES
5 /SMLLATHE,LRGLATHE,CARVER,LABOR/ ;
6
7 PARAMETER PRICE(CHAIRS) PRODUCT PRICES BY PROCESS
8 /FUNCTIONAL 82, FANCY 105/
9 COST(CHAIRS) BASE COST BY CHAIR
10 /FUNCTIONAL 15, FANCY 25/
11 EXTRACOST(CHAIRS,PROCESS) EXTRA COST BY PROCESS
12 / FUNCTIONAL.MXSMLLATHE 1.0 ,
13 FUNCTIONAL.MXLRGLATHE 0.7
14 ,FANCY. MXSMLLATHE 1.5,
15 FANCY. MXLRGLATHE 1.6/
16 RESORAVAIL(RESOURCE) RESOURCE AVAILABILITY
17 /SMLLATHE 140, LRGLATHE 90,
18 CARVER 120, LABOR 125/ ;
19
20 TABLE RESOURUSE(RESOURCE,CHAIRS,PROCESS) RESOURCE USAGE
21 FUNCTIONAL.NORMAL FUNCTIONAL.MXSMLLATHEFUNCTIONAL.MXLRGLATHE
22 SMLLATHE 0.80 1.30 0.20
23 LRGLATHE 0.50 0.20 1.30
24 CARVER 0.40 0.40 0.40
25 LABOR 1.00 1.05 1.10
26 + FANCY.NORMAL FANCY.MXSMLLATHE FANCY.MXLRGLATHE
27 SMLLATHE 1.20 1.70 0.50
28 LRGLATHE 0.70 0.30 1.50
29 CARVER 1.00 1.00 1.00
30 LABOR 0.80 0.82 0.84 ;
31
32 POSITIVE VARIABLES
33 PRODUCTION(CHAIRS,PROCESS) ITEMS PRODUCED BY PROCESS;
34 VARIABLES
35 PROFIT TOTAL PROFIT;
36 EQUATIONS
37 OBJT OBJECTIVE FUNCTION ( PROFIT )
38 AVAILABLE(RESOURCE) RESOURCES AVAILABLE ;
39
40 OBJT.. PROFIT =E= SUM((CHAIRS,PROCESS),
41 (PRICE(CHAIRS)-COST(CHAIRS)-EXTRACOST(CHAIRS,PROCESS))
42 * PRODUCTION(CHAIRS,PROCESS)) ;
43
44 AVAILABLE(RESOURCE)..
45 SUM((CHAIRS,PROCESS),
46 RESOURUSE(RESOURCE,CHAIRS,PROCESS)*PRODUCTION(CHAIRS,PROCESS))
47 =L= RESORAVAIL(RESOURCE);
48
49 MODEL RESALLOC /ALL/;
50 SOLVE RESALLOC USING LP MAXIMIZING PROFIT;

```

**Table 5.5. Optimal Solution to the E-Z Chair Makers Problem**

Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
$X_1$	62.23	0	1	0	33.33
$X_2$	0	-11.30	2	0	25.79
$X_3$	0	-4.08	3	16.91	0
$X_4$	73.02	0	4	0	27.44
$X_5$	0	-8.40			
$X_6$	5.18	0			

**Table 5.6. Formulation of Transportation Example**

Min	$20X_{11} + 40X_{12} + 35X_{13} + 120X_{14}$	$+ 50X_{21} + 60X_{22} + 20X_{23} + 70X_{24} + 90X_{31} + 35X_{32} + 70X_{33} + 40X_{34}$	
s.t.	$X_{11} + X_{12} + X_{13} + X_{14}$	$X_{21} + X_{22} + X_{23} + X_{24}$	# 100
		$X_{31} + X_{32} + X_{33} + X_{34}$	# 75
	$X_{11}$	$+ X_{21} + X_{31}$	# 90
	$X_{12}$	$+ X_{22} + X_{32}$	\$ 30
	$X_{13}$	$+ X_{23} + X_{33}$	\$ 75
	$X_{14}$	$+ X_{24} + X_{34}$	\$ 90
		$X_{11} \leq 0$	\$ 50



**Table 5.7. GAMS Statement of Transportation Example**

```

1 SETS PLANT PLANT LOCATIONS
2 /NEWYORK, CHICAGO, LOSANGLS/
3 MARKET DEMAND MARKETS
4 /MIAMI, HOUSTON, MINEPLIS, PORTLAND/
5
6 PARAMETERS SUPPLY(PLANT) QUANTITY AVAILABLE AT EACH PLANT
7 /NEWYORK 100, CHICAGO 75, LOSANGLS 90/
8 DEMAND(MARKET) QUANTITY REQUIRED BY DEMAND MARKET
9 /MIAMI 30, HOUSTON 75,
10 MINEPLIS 90, PORTLAND 50/;
11
12 TABLE DISTANCE(PLANT,MARKET) DISTANCE FROM EACH PLANT TO EACH MARKET
13
14 MIAMI HOUSTON MINEPLIS PORTLAND
15 NEWYORK 3 7 6 23
16 CHICAGO 9 11 3 13
17 LOSANGLS 17 6 13 7;
18
19
20 PARAMETER COST(PLANT,MARKET) CALCULATED COST OF MOVING GOODS ;
21 COST(PLANT,MARKET) = 5 + 5 * DISTANCE(PLANT,MARKET) ;
22
23 POSITIVE VARIABLES
24 SHIPMENTS(PLANT,MARKET) AMOUNT SHIPPED OVER A TRANSPORT ROUTE;
25 VARIABLES
26 TCOST TOTAL COST OF SHIPPING OVER ALL ROUTES;
27 EQUATIONS
28 TCOSTEQ TOTAL COST ACCOUNTING EQUATION
29 SUPPLYEQ(PLANT) LIMIT ON SUPPLY AVAILABLE AT A PLANT
30 DEMANDEQ(MARKET) MINIMUM REQUIREMENT AT A DEMAND MARKET;
31
32 TCOSTEQ.. TCOST =E= SUM((PLANT,MARKET), SHIPMENTS(PLANT,MARKET)*
33 COST(PLANT,MARKET)) ;
34
35 SUPPLYEQ(PLANT).. SUM(MARKET, SHIPMENTS(PLANT, MARKET)) =L= SUPPLY(PLANT);
36
37 DEMANDEQ(MARKET).. SUM(PLANT, SHIPMENTS(PLANT, MARKET)) =G=
DEMAND(MARKET);
38
39 MODEL TRANSPORT /ALL/;
40 SOLVE TRANSPORT USING LP MINIMIZING TCOST;

```

**Table 5.8. Optimal Solution to the ABC Company Problem**

Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
$X_{11}$	30	0	1	20	0
$X_{12}$	35	0	2	0	-15
$X_{13}$	15	0	3	0	-5
$X_{14}$	0	75	4	0	20
$X_{21}$	0	45	5	0	40
$X_{22}$	0	35	6	0	35
$X_{23}$	75	0	7	0	45
$X_{24}$	0	40			
$X_{31}$	0	75			
$X_{32}$	40	0			
$X_{33}$	0	40			
$X_{34}$	50	0			

**Table 5.9. Optimal Shipping Pattern for the ABC Company**

Origin	Destination							
	Miami		Houston		Minneapolis		Portland	
	Units	Variable	Units	Variable	Units	Variable	Units	Variable
New York	30	$X_{11}$	35	$X_{12}$	15	$X_{13}$		
Chicago					75	$X_{23}$		
Los Angeles			40	$X_{32}$			50	$X_{34}$

**Table 5.10. Optimal Dual Solution to the ABC Company Problem**

Variable	Value	Reduced Cost	Equation	Level	Shadow Price
$U_1$	0	-20	1	-20	30
$U_2$	15	0	2	40	35
$U_3$	5	0	3	35	15
$V_1$	20	0	4	45	0
$V_2$	40	0	5	5	0
$V_3$	35	0	6	25	0
$V_4$	45	0	7	20	75
			8	30	0
			9	15	0
			10	35	40
			11	30	0
			12	40	50

**Table 5.11. Ingredient Costs for Diet Problem Example per kg**

Corn	\$0.133
Dical	\$0.498
Alfalfa hay	\$0.077
Salt	\$0.110
Soybeans	\$0.300
Vitamin A	\$0.286
Urea	\$0.332

**Table 5.12. Required Nutrient Characteristics per Kilogram of Mixed Feed**

Nutrient	Unit	Minimum Amount	Maximum Amount
Net energy	Mega calories	1.34351	--
Digestible protein	Kilograms	0.071	0.13
Fat	Kilograms	--	0.05
Vitamin A	International Units	2200	--
Salt	Kilograms	0.015	0.02
Calcium	Kilograms	0.0025	0.01
Phosphorus	Kilograms	0.0035	0.012
Weight	Kilograms	1	1

**Table 5.13. Nutrient Content per Kilogram of Feeds**

	Dical			Vitamin A	Potato
Net energy	1.48	0.49	1.29		1.39
Digestible protein	0.075	0.127	0.438	2.62	0.032
Fat	0.0357	0.022	0.013		0.009
Vitamin A	600	50880	80	2204600	
Salt				1	
Calcium	0.0002	0.0125	0.0036	0.2313	0.002
Phosphorus	0.0035	0.0023	0.0075	0.68	0.0024

Table 5.14. Primal Formulation of Feed Problem								
	Corn	Hay	Soybean	Urea	Dical	Salt	Vitamin A	Slurry
Min	$.133X_C + .077X_H + .3X_{SB} + .332X_{Ur} + .498X_d + .110X_{SLT} + .286X_{VA} + PX_{SL}$							
s.t.	$.075X_C + .127X_H + .438X_{SB} + 2.62X_{Ur}$							$+ .032X_{SL} \# .13$
Max	$.0357X_C + .022X_H + .013X_{SB}$							$+ .009X_{SL} \# .05$
Nut						$X_{SLT}$		$\# .02$
	$.0002X_C + .0125X_H + .0036X_{SB} + .2313X_d$							$+ .002X_{SL} \# .01$
	$.0035X_C + .0023X_H + .0075X_{SB} + .68X_{Ur} + .1865X_d$							$+ .0024X_{SL} \# .012$
Min	$1.48X_C + .49X_H + 1.29X_{SB}$							$+ 1.39X_{SL} \$ 1.34351$
Nut	$.075X_C + .127X_H + .438X_{SB} + 2.62X_{Ur}$							$+ .032X_{SL} \$ .071$
	$600X_C + 50880X_H + 80X_{SB}$						$+ 2204600X_{VA}$	$\$ 2200$
						$X_{SLT}$		$\$ .015$
	$.0002X_C + .0125X_H + .0036X_{SB} + .2313X_d$							$+ .002X_{SL} \$ .0025$
	$.0035X_C + .0023X_H + .0075X_{SB} + .68X_{Ur} + .1865X_d$							$+ .0024X_{SL} \$ .0035$
Volume	$X_C + X_H + X_{SB} + X_{Ur} + X_d + X_{SLT} + X_{VA} + X_{SL} = 1$							

**Table 5.15 GAMS Formulation of Diet Example**

```

1 SET INGREDT NAMES OF THE AVAILABLE FEED INGREDIENTS
2   /CORN, HAY, SOYBEAN,UREA,DICAL,SALT,VITA,SLURRY/
3   NUTRIENT NUTRIENT REQUIREMENT CATEGORIES
4   /NETENGY, PROTEIN, FAT, VITALIM, SALTLM, CALCIUM, PHOSPHRS/
5   LIMITS TYPES OF LIMITS IMPOSED ON NUTRIENTS /MINIMUM, MAXIMUM/;
6 PARAMETER INGREDCOST(INGREDT) FEED INGREDIENT COSTS PER KG PURCHASED
7   /CORN .133, HAY .077, SOYBEAN .300, UREA .332
8   , DICAL .498,SALT .110, VITA .286, SLURRY .01/;
9 TABLE NUTREQUIRE(NUTRIENT, LIMITS) NUTRIENT REQUIREMENTS
10      MINIMUM      MAXIMUM
11   NETENGY      1.34351
12   PROTEIN      .071      .130
13   FAT          0      .05
14   VITALIM      2200
15   SALTLM       .015      .02
16   CALCIUM      .0025     .0100
17   PHOSPHRS     .0035     .0120;
18 TABLE CONTENT(NUTRIENT, INGREDT) NUTRIENT CONTENTS PER KG OF FEED
19      CORN  HAY  SOYBEAN  UREA  DICAL  SALT  VITA  SLURRY
20  NETENGY      1.48  .49  1.29
21  PROTEIN      .075  .127  .438  2.62
22  FAT          .0357  .022  .013
23  VITALIM      600  50880  80
24  SALTLM
25  CALCIUM      .0002  .0125  .0036      .2313
26  PHOSPHRS     .0035  .0023  .0075  .68  .1865
27
28 POSITIVE VARIABLES
29 FEEDUSE(INGREDT) AMOUNT OF EACH INGREDIENT USED IN MIXING FEED;
30 VARIABLES
31 COST PER KG COST OF THE MIXED FEED;
32 EQUATIONS
33 OBJT      OBJECTIVE FUNCTION ( TOTAL COST OF THE FEED )
34 MAXBD(NUTRIENT) MAXIMUM LIMITS ON EACH NUTRIENT IN THE BLENDED FEED
35 MINBD(NUTRIENT) MINIMUM LIMITS ON EACH NUTRIENT IN THE BLENDED FEED
36 WEIGHT    REQUIREMENT THAT EXACTLY ONE KG OF FEED BE PRODUCED;
37
38 OBJT.. COST =E= SUM(INGREDT,INGREDCOST(INGREDT) * FEEDUSE(INGREDT))
39 MAXBD(NUTRIENT)$NUTREQUIRE(NUTRIENT,"MAXIMUM")..
40   SUM(INGREDT,CONTENT(NUTRIENT , INGREDT) * FEEDUSE(INGREDT))
41   =L= NUTREQUIRE(NUTRIENT, "MAXIMUM");
42 MINBD(NUTRIENT)$NUTREQUIRE(NUTRIENT,"MINIMUM")..
43   SUM(INGREDT,CONTENT(NUTRIENT, INGREDT) * FEEDUSE(INGREDT))
44   =G= NUTREQUIRE(NUTRIENT, "MINIMUM");
45 WEIGHT.. SUM(INGREDT, FEEDUSE(INGREDT)) =E= 1. ;
46 MODEL FEEDING /ALL/;
47 SOLVE FEEDING USING LP MINIMIZING COST;
48
49 SET VARYPRICE PRICE SCENARIOS /1*30/
50 PARAMETER SLURR(VARYPRICE,*)
51 OPTION SOLPRINT = OFF;
52 LOOP (VARYPRICE,
53   INGREDCOST("SLURRY")= 0.01 + (ORD(VARYPRICE)-1)*0.005;
54   SOLVE FEEDING USING LP MINIMIZING COST;
55   SLURR(VARYPRICE,"SLURRY") = FEEDUSE.L("SLURRY");
56   SLURR(VARYPRICE,"PRICE") = INGREDCOST("SLURRY") ) ;
57   DISPLAY SLURR;

```



**Table 5.16. Optimal Primal Solution to the Diet Example Problem**

Variable	Value	Reduced Cost	Equation	Slack	Price
X <sub>C</sub>	0	0.095	Protein L Max	0.059	0
X <sub>H</sub>	0.001	0	Fat Max	0.041	0
X <sub>SB</sub>	0.011	0	Salt Max	0.005	0
X <sub>Ur</sub>	0.014	0	Calcium Max	0.007	0
X <sub>d</sub>	0.002	0	Phosphrs	0.000	-2.207
X <sub>SLT</sub>	0.015	0	Net Engy Min	0.000	0.065
X <sub>VA</sub>	0.001	0	Protein Min	0.000	0.741
X <sub>SL</sub>	0.956	0	Vita Lim Min	0.000	0
			Salt Lim Min	0.000	0.218
			Calcium Min	.000	4.400
			Phosphrs	0.008	0
			Weight	0.000	-0.108

**Table 5.17. Dual Formulation of Feed Mix Example Problem**

	( <sub>1</sub>	( <sub>2</sub>	( <sub>3</sub>	( <sub>4</sub>	( <sub>5</sub>	\$ <sub>1</sub>	\$ <sub>2</sub>	\$ <sub>3</sub>	\$ <sub>4</sub>	\$ <sub>5</sub>	\$ <sub>6</sub>	"
Max	-.13	-.05	-.02	-.01	-.12	+ 1.34351	+ .071	+ 2200	+ .015	+ .0025	+ .0035	+ 1
	-.075	-.0357		-.00	-.0035	+ 1.48	+ .075	+ 600		+ .0002	+ .0035	+ 1 # .133
	-.127	-.022		-.01	-.0023	+ .49	+ .127	+ 50880		+ .0125	+ .0023	+ 1 # .077
	-.438	-.013		-.00	-.0075	+ 1.29	+ .438	+ 80		+ .0036	+ .0075	+ 1 # .3
	-2.62				-.68		+ 2.62				+ .68	+ 1 # .332
				-.23	-.1865					+ .2313	+ .1865	+ 1 # .498
		- 1							+ 1			+ 1 # .110
							+ 220460					+ 1 # .286
	-.032	-.009		-.00	-.0024	+ 1.39	+ .032		+ .002	+ .0024	+ 1	# P

**Table 5.18. Optimal Solution to the Dual of the Feed Mix Problem**

Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
( <sub>1</sub> )	0	-0.059	Corn	0.095	0
( <sub>2</sub> )	0	-0.041	Hay	0	0.001
( <sub>3</sub> )	0	-0.005	Soybean	0	0.011
( <sub>4</sub> )	0	-0.007	Urea	0	0.014
( <sub>5</sub> )	2.207	0	Dical	0	0.002
\$ <sub>1</sub>	0.065	0	Salt	0	0.015
\$ <sub>2</sub>	0.741	0	Vita	0	0.001
\$ <sub>3</sub>	0	0	Slurry	0	0.956
\$ <sub>4</sub>	0.218	0			
\$ <sub>5</sub>	4.400	0			
\$ <sub>6</sub>	0	-0.008			
"	-0.108	0			

**Table 5.19. Data for the Wheat and Straw Example Problem**

	Outputs and Inputs Per Acre							
Wheat yield in bu.	30	50	65	75	80	80	75	
Wheat straw yield/bales	10	17	22	26	29	31	32	
Fertilizer use in Kg.	0	5	10	15	20	25	30	
Seed in pounds	10	10	10	10	10	10	10	



**Table 5.20. GAMS Formulation of the Joint Products Example**

```

1 SET PRODUCTS LIST OF ALTERNATIVE PRODUCT /WHEAT, STRAW/
2 INPUTS PURCHASED INPUTS /SEED, FERT/
3 FIXED FIXED INPUTS /LAND/
4 PROCESS POSSIBLE INPUT COMBINATIONS /Y1*Y7/;
5
6 PARAMETER PRICE(PRODUCTS) PRODUCT PRICES /WHEAT 4.00, STRAW 0.50/
7 COST(INPUTS) INPUT PRICES /SEED 0.20, FERT 2.00/
8 PRODCOST(PROCESS) PRODUCTION COSTS BY PROCESS
9 AVAILABLE(FIXED) FIXED INPUTS AVAILABLE / LAND 500 /;
10
11 PRODCOST(PROCESS) = 5;
12
13 TABLE YIELDS(PRODUCTS, PROCESS) YIELDS FROM THE PRODUCTION
POSSIBILITIES
14 Y 1 Y 2 Y 3 Y 4 Y 5 Y 6 Y 7
15 WHEAT 30 50 65 75 80 80 75
16 STRAW 10 17 22 26 29 31 32;
17
18 TABLE USAGE(INPUTS,PROCESS) PURCHASED INPUT USAGE BY PRODUCTION
POSSIBILITIES
19 Y 1 Y 2 Y 3 Y 4 Y 5 Y 6 Y 7
20 SEED 10 10 10 10 10 10 10
21 FERT 0 5 10 15 20 25 30;
22
23 TABLE FIXUSAGE(FIXED,PROCESS) FIXED INPUT USAGE BY PRODUCTION
POSSIBILITIES
24 Y 1 Y 2 Y 3 Y 4 Y 5 Y 6 Y 7
25 LAND 1 1 1 1 1 1 1;
26
27 POSITIVE VARIABLES
28 SALES(PRODUCTS) AMOUNT OF EACH PRODUCT SOLD
29 PRODUCTION(PROCESS) LAND AREA GROWN WITH EACH INPUT
PATTERN
30 BUY(INPUTS) AMOUNT OF EACH INPUT PURCHASED ;
31 VARIABLES
32 NETINCOME NET REVENUE (PROFIT);
33 EQUATIONS
34 OBJT OBJECTIVE FUNCTION (NET REVENUE)
35 YELDBAL(PRODUCTS) BALANCES PRODUCT SALE WITH PRODUCTION
36 INPUTBAL(INPUTS) BALANCE INPUT PURCHASES WITH USAGE
37 AVAIL(FIXED) FIXED INPUT AVAILABILITY;
38
39 OBJT.. NETINCOME =E=
40 SUM(PRODUCTS , PRICE(PRODUCTS) * SALES(PRODUCTS))
41 - SUM(PROCESS , PRODCOST(PROCESS) *
PRODUCTION(PROCESS))
42 - SUM(INPUTS , COST(INPUTS) * BUY(INPUTS));
43 YELDBAL(PRODUCTS)..
44 SUM(PROCESS, YIELDS(PRODUCTS,PROCESS) * PRODUCTION(PROCESS))
45 =G= SALES(PRODUCTS);
46 INPUTBAL(INPUTS)..
47 SUM(PROCESS, USAGE(INPUTS,PROCESS) * PRODUCTION(PROCESS))
48 =L= BUY(INPUTS);
49 AVAIL(FIXED)..
50 SUM(PROCESS, FIXUSAGE(FIXED,PROCESS)*PRODUCTION(PROCESS))
51 =L= AVAILABLE(FIXED);
52
53 MODEL JOINT /ALL/;

```



**Table 5.21. Optimal Solution of the Wheat and Straw Example Problem**

Objective Function Value 143,750					
Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
X <sub>1</sub>	40,000	0	Wheat	0	-4
X <sub>2</sub>	14,500	0	Straw	0	-0.5
Y <sub>1</sub>	0	-169.50	Fertilizer	0	2
Y <sub>2</sub>	0	-96.00	Seed	0	0.2
Y <sub>3</sub>	0	-43.50	Land	0	287.5
Y <sub>4</sub>	0	-11.50			
Y <sub>5</sub>	500	0			
Y <sub>6</sub>	0	-9.00			
Y <sub>7</sub>	0	-38.50			
Z <sub>1</sub>	10,000	0			
Z <sub>2</sub>	5,000	0			

**Table 5.22. Dual Formulation of Example Joint Products Problem**

Min					500U
s.t.	V <sub>1</sub>				\$ 4
		V <sub>2</sub>			\$ .5
	-30V <sub>1</sub>	- 10V <sub>2</sub>	+ 10W <sub>2</sub>	+ U	\$ -5
	-50V <sub>1</sub>	- 17V <sub>2</sub>	+ 5W <sub>1</sub>	+ 10W <sub>2</sub>	+ U \$ -5
	-65V <sub>1</sub>	- 22V <sub>2</sub>	+ 10W <sub>1</sub>	+ 10W <sub>2</sub>	+ U \$ -5
	-75V <sub>1</sub>	- 26V <sub>2</sub>	+ 15W <sub>1</sub>	+ 10W <sub>2</sub>	+ U \$ -5
	-80V <sub>1</sub>	- 29V <sub>2</sub>	+ 20W <sub>1</sub>	+ 10W <sub>2</sub>	+ U \$ -5
	-80V <sub>1</sub>	- 31V <sub>2</sub>	+ 25W <sub>1</sub>	+ 10W <sub>2</sub>	+ U \$ -5
	-75V <sub>1</sub>	- 32V <sub>2</sub>	+ 30W <sub>1</sub>	+ 10W <sub>2</sub>	+ U \$ -5
			-W <sub>1</sub>		\$ -2
				-W <sub>2</sub>	\$ -0.2
	V <sub>1</sub>	, V <sub>2</sub>	, W <sub>1</sub>	, W <sub>2</sub>	, U \$ 0

**Table 5.23. Optimal Dual Solution of the Wheat and Straw**

Objective Function Value 143,750					
Variable	Value	Reduced Cost	Equation	Slacks	Shadow
V <sub>1</sub>	4	0	Wheat	0	40,000
V <sub>2</sub>	0.5	0	Straw	0	14,500
W <sub>1</sub>	2	0	Prod 1	169.5	0
W <sub>2</sub>	0.2	0	Prod 2	96	0
U	287.5	0	Prod 3	43.5	0
			Prod 4	11.5	0
			Prod 5	0	500
			Prod 6	9	0
			Prod 7	38.5	0
			Fertilizer	0	10,000
			Seed	0	5,000

**Table 5.24 Alternative Computer Inputs for a Model**

Simple GAMS Input File

```

POSITIVE VARIABLES  X1 ,  X2,  X3
VARIABLES           Z
EQUATIONS           OBJ,  CONSTRAIN1 ,  CONSTRAIN2;
OBJ..               Z =E= 3 * X1 + 2 * X2 + 0.5* X3;
CONSTRAIN1..       X1 + X2 +X3=L= 10;
CONSTRAIN2..       X1 - X2 =L= 3;
MODEL PROBLEM /ALL/;
SOLVE PROBLEM USING LP MAXIMIZING Z;
    
```

LINDO Input file

```

MAX 3 * X1 + 2 * X2 + 0.5* X3;
ST
X1 + X2 +X3 < 10
X1 - X2 < 3
END
GO
    
```

Tableau Input File

```

5      3
3.    2.    0.5    0.    0.
1.    1.    1.    1.    0.    10.
1.   -1.    0.    0.    1.    3.
    
```

MPS Input File

```

NAME           CH2MPS
ROWS
N              R 1
L              R 2
L              R 3
COLUMNS
X 1            R 0           3.           R 1           1.
X 1            R 3           1.
X 2            R 0           2.           R 1           1.
X 2            R 1           -1.
X 3            R 0           0.5          R 1           1.
RHS
RHS 1          R 1           10.          R 1           3.
ENDDATA
    
```

More Complex GAMS input file

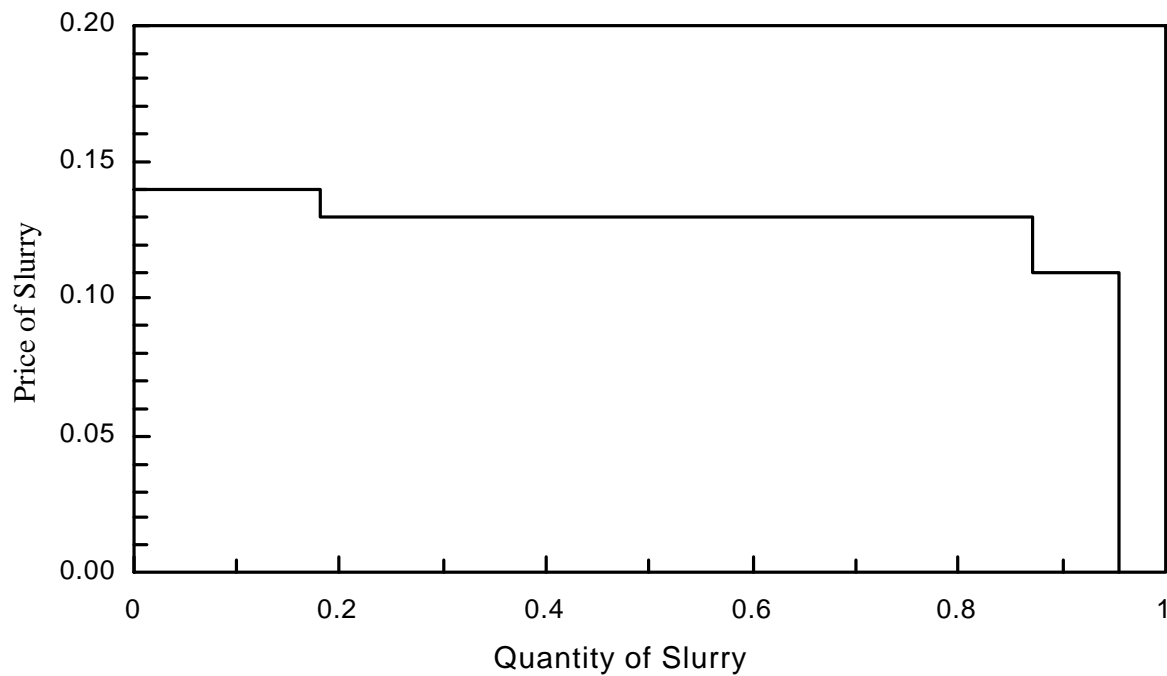
```

SET  PROCESS      TYPES OF PRODUCTION PROCESSES  /X1,X2,X3/
     RESOURCE     TYPES OF RESOURCES              /CONSTRAIN1,CONSTRAIN2/
PARAMETER
PRICE(PROCESS)   PRODUCT PRICES BY PROCESS /X1 3,X2 2,X3 0.5/
PRODCOST(PROCESS)  COST BY PROCESS           /X1 0 ,X2 0, X3 0/
RESORAVAIL(RESOURCE)  RESOURCE AVAILABILITY  /CONSTRAIN1 10 ,CONSTRAIN2 3/
TABLE RESOURUSE(RESOURCE,PROCESS) RESOURCE USAGE
           X 1           X 2           X 3
CONSTRAIN1  1           1           1
CONSTRAIN2  1           -1
POSITIVE VARIABLES  PRODUCTION(PROCESS) ITEMS PRODUCED BY PROCESS;
VARIABLES           PROFIT                TOTALPROFIT;
EQUATIONS           OBJT                    OBJECTIVE FUNCTION ( PROFIT )
                   AVAILABLE(RESOURCE) RESOURCES AVAILABLE ;
    
```

```
OBJT.. PROFIT=E= SUM(PROCESS,(PRICE(PROCESS)-PRODCOST(PROCESS))*  
                PRODUCTION(PROCESS)) ;  
AVAILABLE(RESOURCE).. SUM(PROCESS,RESOURUSE(RESOURCE,PROCESS)  
                *PRODUCTION(PROCESS)) =L= RESORAVAIL(RESOURCE);  
MODEL RESALLOC /ALL/;  
SOLVE RESALLOC USING LP MAXIMIZING PROFIT;
```

---

**Figure 5.1 Demand Schedule for Potato Slurry in Feed Mix Example**



## CHAPTER VI: TOWARD PROPER MODELING

There is considerable role for judgment when modeling and developing data. The applied modeler must make assumptions regarding the variables, constraints, and coefficients. These assumptions determine model performance and usefulness.

In this chapter the identification of structural components and the development of data are discussed. The material presented here is reinforced by material in subsequent chapters. References are made to this later material, and readers may wish to consult it for more detailed explanations.

Before beginning this section, the authors must acknowledge their debt to Heady and Candler's "Setting Up Linear Programming Models" chapter and conversations with Wilfred Candler.

### 6.1 Structural Component Identification

The LP problem can be expressed as

$$\begin{array}{ll} \text{Max} & CX \\ \text{s.t.} & AX \# b \\ & X \geq 0 \end{array}$$

In order to formulate an applied LP problem, one must identify the constraints, variables and relevant numerical parameter values.

#### 6.1.1 Development of Model Constraints

Heady and Candler categorize LP constraints as technical, institutional, and subjective.

Constraints also arise because of convenience or model formulation requirements. Technical constraints depict limited resources, intermediate products, or contractual requirements. Technical constraints also express complementary, supplementary, and competitive relationships among variables. Collectively, the technical constraints define the production possibilities and provide links between variables. Institutional constraints reflect external regulations imposed on the problem. Examples include credit limits or farm program participation requirements. Subjective constraints are imposed by the decision maker or modeler. These might include a hired labor limitation based on the decision maker's willingness to supervise labor. Convenience constraints facilitate model interpretation and may be included to sum items of interest. Model formulation constraints aid in problem depiction. These include constraints used in conjunction with approximations. Within and across these groupings, constraints can take on a number of different forms. A more extensive definition of these forms is presented in the LP Modeling Summary chapter.

Generally, the constraints included should meaningfully limit the decision variables. The modeler should begin by defining constraint relations for those production resources and commitments which limit production or are likely to do so. This involves consideration of the timing of resource availability. Often, problems covering seasonal production or utilizing seasonally fluctuating resources will contain time desegregated constraints. Heady and Candler argue that multiple constraints are needed to depict availability of a resource whenever the marginal rate of factor substitution between resource usage in different time periods does not equal one. Constraints must be developed so that the resources available within a particular constraint are freely substitutable. Cases of imperfect substitution will require multiple constraints.

Two other points should be made regarding constraint definition. First, an LP solution will include no more variables at a nonzero level than the number of constraints (including the number of upper and lower bounds). Thus, the number of constraints directly influences the number of nonzero variables in the



optimal solution. However, one should not simply define additional constraints as: 1) this usually results in additional nonzero slack variables without substantially altering the solution; and 2) one must not impose nonsensical constraints.

Second, subjective constraints should not be imposed before determining their necessity. Often, subjective constraints "correct" model deficiencies. But the cause of these deficiencies is frequently missing either technical constraints or omitted variables. For example, models often yield excessively specialized solutions which force variables into the solution. This is often combated by imposing "flexibility" constraints as suggested by Day (1963), or discussed in Sengupta and Sfeir. Often, however, the real deficiency may be the depiction of the time availability of resources (Baker and McCarl). In such a case, the subjective constraints give an inadequate model a "nominal" appearance of reality, but are actually causing the "right" solution to be observed for the wrong reason.

### **6.1.2 Avoiding Improper Constraint Specifications**

LP model constraints have higher precedence than the objective function. The first major effort by any LP solver is the discovery of a feasible solution. The solver then optimizes within the feasible region. This has several implications for identification and specification of constraints.

First, the modeler must question whether a constraint should be established so it always restricts the values of the decision variables. Often, it may be desirable to relax a constraint allowing resource purchases if the value of a resource becomes excessively high.

Second, modelers should be careful in the usage of minimum requirement constraints (e.g.,  $X_1 + X_2 \geq 10$ ). Minimum requirements must be met before profit seeking production can proceed. Often purchase variables should be entered to allow buying items to meet the requirements.

Third, judicious use should be made of equality constraints. Modelers should use the weakest form of a constraint possible. Consider the following example:

$$\begin{array}{rcl}
 \text{Max} & 3X & + 2Y \\
 \text{s.t.} & X & + Y \text{ ? } 0 \\
 & X & \# 10 \\
 & & Y \# 15
 \end{array}$$

Where ? is the constraint type (either = or # ), X depicts sales and Y production? Further, suppose we have made a mistake and have specified the cost of production as a revenue item (i.e., the +2Y should be -2Y). Now, if the relation is an equality, then the optimal solution is  $X = Y = 10$  (see file SIXEQ), and we do not discover the error (although the dual variable on the first constraint is -2). On the other hand, if the relation is # then we would produce  $Y = 15$  units while selling only  $X = 10$  units (see file SIXLT). Thus, the weaker inequality form of the constraint allows an unrealistic production pattern indicating that something is wrong with the model.

### 6.1.3 Variable Identification

LP variables are the unknowns of the problem. Variables are included for either technical, accounting or convenience reasons. Technical variables change value in response to the objective function and constraints. Convenience variables may not always respond to the objective function. Rather, they may be constrained at certain levels. These might include variables representing the number of acres of land used for houses and buildings. Accounting variables facilitate solution summarization and model use.

It is critically important that the technical variables logically respond to the objective function within the range of values imposed by the constraints. For example, one could setup a farm problem with variables responding to an objective of minimizing soil erosion. However, farmers choosing acreage may not primarily try to minimize erosion; most farmers are also profit oriented.

Many types of technical variables are possible. A taxonomy is discussed in the LP Modeling Summary chapter.

Variables must be in consistent units. Actually, there are no strict LP requirements on the variable

units. However, the intersection of the variable and constraint units impose requirements on the  $a_{ij}$ 's as discussed below. Now, when can multiple variables be handled as one variable and when can't they?

There are several cases when multiple variables must be defined:

- (a) When more than one process can be used to produce the same output using different resource mixes; e.g., the production of an item using either of two different machines.
- (b) When different processes produce different outputs using common resources; i.e., one can use essentially the same resources to produce either 2 x 4 or 4 x 4 sawn lumber.
- (c) When products can be used in several ways; e.g., selling chickens that can be quartered or halved.

Collectively, different variables should be used where their coefficients differ (i.e., the objective function or  $a_{ij}$  coefficients differ across production possibilities). However, the coefficients should not be strictly proportional (i.e., one variable having twice the objective function value of another while using twice the resources).

Criteria may also be developed where two variables may be treated as one. The simplest case occurs when the coefficients of one variable are simple multiples of another ( $a_{ij} = K a_{im}$  and  $c_j = K c_m$ ). The second case occurs when one variable uniquely determines another; i.e., when  $n$  units of the first variable always implies exactly  $m$  units of the second.

#### **6.1.4 Objective Function**

Once the variables and constraints have been delineated, then the objective function must be specified. The variables and constraints jointly define the feasible region. However, the objective function identifies the "optimal" point. Thus, even with the proper variables and constraints, the solution is only as good as the objective function. Ordinarily, the first objective function specification is inadequate. Most situations do not involve strict profit maximization, but also may involve such things as risk avoidance or labor/leisure tradeoffs.

Multiple objective models are discussed in the multi-objective and risk chapters. Also, ranging analysis can be used to discover whether the solution will change with alterations in the objective function.

### **6.1.5 Development of Model Structure**

Model definition is an iterative process. Consider a simple example where a profit maximizing firm produces four crops subject to land and labor limitations. Suppose that the crops are grown at different times of the year. Crop 1 is planted in the spring and harvested in the summer; crops 2 and 3 are planted in the spring and harvested in the fall; and crop 4 is planted following crop 1 and is harvested in the fall.

The first step in developing a model is to lay out a table with potential variables across the top and constraints/objective function down the side. In this case we start with the layout in Table 6.1 where the variables are crop acreages and the constraints are land and labor availability. We then begin to define coefficients. Suppose  $c_i$  gives the gross profit margins for crop  $i$ . Simultaneously, land use coefficients and the land endowment ( $L$ ) are entered. However, the land constraint only has entries for crops 1, 2 and 3, as crop 4 uses the same land as crop 1. Thus, a single land constraint restricts land use on an annual basis. We also need a constraint which links land use by crop 4 to the land use by crop 1. Thus, our formulation is altered as shown in Table 6.2, where the second constraint imposes this linkage.

Now we turn our attention to labor. In this problem, labor is not fully substitutable between all periods of the year, i.e., the elasticity of substitution criterion is not satisfied. Thus, we must develop time-specific labor constraints for spring, summer and fall. The resultant model is shown in Table 6.3. Subsequently, we would fill in the exact labor coefficients; i.e.; the  $d$ 's and right hand sides.

This iterative process shows how one might go about defining the rows and columns. In addition, one could further disaggregate the activities to allow for different timing possibilities. For example, if Crop 1 produced different yields in different spring planting periods, then additional variables and constraints would need to be defined.

## 6.2 The Proper Usage of Variables and Constraints

Students often have difficulties with the definition of variables and constraints. This section is intended to provide insight by presenting a number of proper and improper examples.

The applied LP modeler needs to recognize three concepts when forming constraints and variables. First, the coefficients associated with a variable reflect a simultaneous set of commitments which must occur when a variable is nonzero. All the resources used by a variable must be used to produce its output. Thus, if a variable depicts cattle and calf production using inputs of land, labor, and feed; then the model will simultaneously commit land, labor, and feed in order to get the simultaneous outputs - cattle and calves. One cannot obtain calves without obtaining cattle nor can one obtain cattle and calves without using land, labor, and feed.

Second, the choice is always modeled across variables, never within a variable. For example, suppose there are two ways of producing cattle and calves. These production alternatives would be depicted by two variables, each representing a simultaneous event. The model would reflect choice regarding cattle/calf production within the constraints. These choices do not have to be mutually exclusive; the model may include complementary relationships between variables as well as substitution relationships (i.e. the constraint  $X - Y = 0$  makes X and Y complementary).

Third, resources within a constraint are assumed to be homogeneous commodities. Suppose there is a single constraint for calves with the calves being produced by two variables. In turn, suppose calves may be used in two feeding alternatives. In such a case the calves are treated as perfect substitutes in the feeding processes regardless of how they were produced.

While obvious, it is surprising the number of times there are difficulties with these topics (even with experienced modelers). Thus, we will present cases wherein such difficulties could be encountered.

### 6.2.1 Improper and Proper Handling of Joint Products

Joint products are common in LP formulations. For purposes of illustration, we adopt the following

simplified example.<sup>4</sup> Suppose a chicken is purchased and cut up into four component parts - breasts, legs, necks, and giblets - following the breakdown data in Table 6.4. Also, assume that each chicken weighs three pounds and that there are 1,500 chickens available.

Now suppose that we formulate a profit maximizing LP model. Such a LP model would involve variables for cutting the chickens along with variables for parts sale. Two alternative formulations will be presented: one proper and one improper. These formulations are shown in Table 6.5 and are labeled Formulation 6.5(a) and Formulation 6.5(b).

The models maximize the value of the chicken parts produced. A constraint is needed which limits the number of chickens cut up to the number of chickens available. In both formulations the coefficient 1/3 in the last constraint transforms the chickens disassembled into pounds of chicken rather than whole chickens, so the units of the first variable are in pounds of chicken cut up. The next four variables are the quantities (pounds) of parts sold. In formulation 6.5(a) the constraint labeled Balance restrains the amount sold to the quantity of chicken cut. The formulation maximizes the value of chicken parts sold. The decision is constrained by the quantity of chicken disassembled and chickens available. In Formulation 6.5(b), the objective function and last constraint are the same. However, there are balances for each part.

Now which formulation is improper? Suppose we examine what happens when Y equals one (i.e., that we have acquired one pound of chicken for cutting up). Formulation 6.5(a) implies that variable  $X_1$  could equal two if the other variables were set to zero. Thus, from one pound of chicken two pounds of chicken breasts could be sold. This is not possible. Similarly, 3.43 pounds of legs ( $X_2$ ) could be sold per pound of chicken. 10 pounds of necks ( $X_3$ ) or 20 pounds of giblets ( $X_4$ ) could be sold. In formulation 6.5(b), the acquisition of one pound of chicken would allow only .5 pounds of breasts, .35 pounds of legs and thighs, .1 pounds of necks, and

---

<sup>4</sup>The example is a disassembly problem, which is discussed in the More LP Modeling chapter. Readers having difficulty with its basic structure may wish to study that section.

.05 pounds of giblets.

Clearly, formulation 6.5(b) is the proper formulation. Formulation 6.5(a) depicts improper representation of the joint products allowing an improper choice between the use of all the chicken meat among any of the four component parts. In fact, its optimal solution indicates 90,000 lbs of Giblets can be sold from the 4,500 lbs of chicken (see the file SIX5A on the disk). The component parts are a joint product that should simultaneously occur in the model.

### **6.2.2 Alternatives for the Use of a Product**

Errors also occur when modeling different ways products can be used. Suppose we introduce the option of selling chicken parts or deboning the parts then selling chicken meat. Assume that there are no additional resources involved, and that the meat yields are those in Table 6.4. Again, we will illustrate proper modeling with a right and a wrong formulation.

The first model Table 6.6(a) has three new variables and constraints. The three new variables sell meat at \$1.20. The three new constraints balance meat yields with sale. Thus, the coefficient in the breast quarter meat row is the meat yielded when breast quarter is deboned (the breast quarter poundage per chicken times the percentage of meat in a breast quarter).

Formulation 6.6(b) adds four variables and one row. The first three variables transform each of the products into meat with the fourth selling the resultant meat. The new constant balances the amount of meat yielded with the amount of meat sold.

Now which formulation is proper? Let us examine the implications of setting the variable  $Y$  equal to 1 in Table 6.6(a). As in our earlier discussion the solution would have, variables  $X_1$  through  $X_4$  at a nonzero level. However, in this formulation  $M_1$ ,  $M_2$ , and  $M_3$  would also be nonzero. Since both the  $X$  variables and the  $M$  variables are nonzero, the chicken is sold twice. In Table 6.6(b), when  $Y$  is set to one, then either  $X_1$  or  $M_1$  can be set to .5, but not both (in fact, the sum of  $X_1 + M_1$  can be no greater than 0.5). Thus, the chicken parts

can only be sold once.

Formulation 6.6(b) is proper. Formulation 6.6(a) contains an improper joint product specification, as it simultaneously allocates products to mutually exclusive uses. Formulation 6.6(b) restricts any single part to one use.

### **6.2.3 Improper Specification of Multiple Factor Relationships**

Factor usage is often subject to misspecification in terms of multiple factor relationships. This case is illustrated with yet another extension of the chicken example. We now wish to allow sales of a mixed quarter pack which is composed of an arbitrary combination of breast and leg quarters. Let us introduce two models. The first model has the same constraints as Formulation 6.6(b) but introduces new variables where the breast and leg quarters are put into the mixed quarter package (Formulation 6.7(a)).

Formulation 6.7(b) involves three new variables and one new constraint. The first two variables are the poundage of breast and leg quarters utilized in the mixed packs. The third variable is total poundage of mixed quarter pack sold. The new constraint balances the total poundage of the mixed quarter packs sold with that produced.

Now the question again becomes which is right? Formulation 6.7(a) is improper; the formulation requires that in order to sell one pound of the mixed quarter pack, two pounds of quarters, one of each type, must be committed and leads to a solution where no packs are made (see the file SIX7A). In Formulation 6.7(b) the two sources of quarters are used as perfect substitutes in the quarter pack, permitting any proportion that maximizes profits. The optimal solution shows all leg quarters sold as mixed quarter packs. Formulation 6.7(a) illustrates a common improper specification - requiring that the factors to be used simultaneously when multiple factors may be traded off. One should not require simultaneous factor use unless it is always required. Multiple variables are required to depict factor usage tradeoffs.

### **6.2.4 Erroneous Imperfect Substitution**



Resource substitution may also be incorrectly prevented. Consider a problem depicting regular and overtime labor. Suppose the basic product is chairs where: a) a chair requires 10 hours of labor of which an average of 3 hours comes from overtime labor; b) the firm has an endowment of 77 hours of regular labor at \$10 per hour and up to 27 hours of overtime labor at \$15 per hour. We again introduce two formulations.

In Formulation 6.8(a) the variables indicate the number of chairs to produce and sell, along with the amount of labor to acquire. The constraints give a balance between the chairs produced and sold; balances between the labor/quantities hired versus used; and limits on labor time available.

Model 6.8(b) is essentially the same, however, we have aggregated our labor use-hired balance so that there is no distinction made between the time when labor is used (regular or overtime).

Which formulation is right? This depends on the situation. Suppose that labor works with equal efficiency in both time classes. Thus, one would be technically indifferent to the source of labor although economically the timing has different implications. Now let us examine the formulations by setting  $X_1$  to one. In 6.8(a) the model hires both classes of labor. However, in 6.8(b) only regular time labor would be hired. In fact, in 6.8(a) the overtime limit is the binding constraints and not all regular time labor can be used and only nine chairs are made; whereas in 6.8(b) eleven chairs could be produced and all the labor is used. The second model is the correct one since it makes no technical differentiation between labor sources.

### 6.3 Simple Structural Checking

There are some simple yet powerful techniques for checking LP formulations. Two are discussed here another in Chapter 17.

#### 6.3.1 Homogeneity of Units

There are several general requirements for coefficient units. Consider the LP problem:

$$\begin{array}{ll}
 \text{Max} & c_1X_1 \quad \% \quad c_2X_2 \\
 \text{s.t.} & a_{11}X_1 \quad \% \quad a_{12}X_2 \quad \# \quad b_1 \\
 & a_{21}X_1 \quad \% \quad a_{22}X_2 \quad \# \quad b_2
 \end{array}$$

Suppose that the objective function unit is dollars. Let the first row be a land constraint in acres. Let the second row be a labor constraint in the unit hours. Further, suppose that  $X_1$  represents acres of wheat and  $X_2$  number of beef animals.

What implications do these specifications have for the units within the model? Parameter  $c_1$  must be the dollars earned per acre of wheat while  $c_2$  must be the dollars earned per beef animal. Multiplying these two parameters by the solution values of  $X_1$  and  $X_2$  results in the unit dollars. In turn,  $a_{11}$  represents the acres of land used per acre of wheat. The parameter  $a_{12}$  would be the number of acres of land utilized per beef animal. The units of the right hand side ( $b_1$ ) must be acres. The units of the parameters  $a_{21}$  and  $a_{22}$  would respectively be labor hours utilized per wheat acre and labor hours utilized per beef animal. The units of the right hand side ( $b_2$ ) must be hours of labor.

This example gives a hint of several general statements about units. First, the numerator unit of each coefficient in an equation must be the same and must equal the right-hand side unit. Thus,  $a_{11}$  is the acres of land used per acre of wheat,  $a_{12}$  is the acres of land used per beef animal and  $b_1$  the acres of land available.

Similarly, the coefficients associated with any particular variable must have a common denominator unit, although the numerator will vary. Thus,  $c_1$  is in the units dollars per acre of wheat,  $a_{11}$  is acres of land per acre of wheat, and  $a_{21}$  is the hours of labor per acre of wheat. In addition, note that the units of the decision variable  $X_1$  are acres of wheat. The denominator unit of all coefficients within a column must be the same as the unit of the associated decision variable.

### 6.3.2 Numerical Model Analysis

Another possible type of model analysis involves numerical investigation of the model. Here, one simply mentally fixes variables at certain levels such as the level of 1, and then examines the relationship of this variable with other variables by examining the equations. Examples of this procedure are given in the proper usage section above.

Numerical debugging can also be carried out by making sure that units are proper, and it is possible to utilize all resources and produce all products. Finally, solvers such as OSL contain reduction procedures.

## **6.4 Data Development**

Model specification requires data. The data need to be found, calculated, and checked for consistency. Data development usually takes more time than either model formulation or solution. However, this time is essential. Good solutions do not arise from bad data.

Data development involves a number of key considerations. These include time frame, uncertainty, data sources, consistency, calculation methods, and component specification.

### **6.4.1 Time Frame**

Models must be established with a time frame in mind. The time frame defines the characteristics of the data used. The objective function, technical coefficient ( $a_{ij}$ 's) and right hand side data must be mutually consistent. When the model depicts resource availability on an annual basis, then the objective function coefficients should represent the costs and revenues accruing during that year. A common misspecification involves an annual model containing investment activities with the full investment cost in the objective function.

Dynamic considerations may be relevant in the computation of objective function coefficients. It is crucial that the objective function coefficients be derived in a consistent manner. Returns today and returns in ten years should not be added together on an equal basis. Issues of dynamics and discounting must be considered as discussed in the Dynamic LP Chapter.

### **6.4.2 Uncertainty**

The data developer must consider uncertainty. Coefficients will virtually never be known with certainty. For example, when variables involve transport of goods from one place to another, the transport costs are not entirely certain due to difficulties with pilferage, spoilage, adherence to shipping containers, and leakage.

The modeler is forever facing decisions on how to incorporate data uncertainty. The risk programming chapter presents formal methods for incorporating uncertainty. However, many modelers use average values or conservative estimates.

### **6.4.3 Data Sources**

Data may be developed through statistical estimation or deductive processes. Data for coefficient estimation can be from either cross-sectional or time series sources. Data may be developed using a case firm (or firms) approach where a deductive, economic engineering process is used to manufacture representative coefficient values. Data sources will vary by problem, and the modeler must apply ingenuity as well as problem-specific knowledge to develop consistent, reliable data.

### **6.4.4 Calculation Methods**

Data can be calculated via economic engineering or via statistical methods. While these are only two extremes of a continuum of possibilities, we will discuss only these two. Economic engineering refers to coefficient construction through a deductive approach. For example, suppose we compute the profit contribution of a variable by calculating the per unit yield times sale price less the per acre input usage times input price (i.e., if wheat production yields 40 bushels of wheat which sells for \$5 per bushel and 20 bales of straw each worth \$.50 while input usage is \$30 worth for seed and 6 sacks of fertilizer, which cost \$4 each; then, the objective function coefficient would be \$156.)

At the other extreme, one could develop multiple observations from time series, cross-sectional or subjective sources and use averages, regression or other data summarization techniques. Such data might in turn be transformed using an economic engineering approach to generate relevant coefficients. For example, one might estimate a function statistically relating yield to fertilizer use and labor use. Then one might set a level of fertilizer use, calculate the yield, and use an economic engineering approach to develop the objective function coefficients.

### **6.4.5 Consistency**

Coefficients in a model must be mutually consistent. The most common causes of inconsistency are dynamic inconsistencies and inconsistencies in coefficient units (e.g., a technical coefficient in hours and a right-hand side in thousands of hours). The homogeneity of units rules above must be followed.

### **6.4.6 Specification of Individual Components**

LP problems require right hand side, objective function, and technical coefficient specification. There are comments that can be made pertinent to the specification of each.

#### 6.4.6.1 Objective Function Coefficients

Ordinarily, the objective function coefficients should be the value that the decision maker expects. This is particularly important when using time series data as the decision maker will not necessarily expect the series average. Rather, some extrapolation of the trend may be appropriate. Brink and McCarl encountered difficulties when attempting to validate a LP model because of differences in expectations between the time the model was developed and the time actual decisions were made.

Several other comments are relevant regarding the objective function. First, multiplication of a LP objective function by a positive constant always leads to the same solution in terms of the decision variables. Thus, one does not need to be extremely concerned about the absolute magnitude of the objective function coefficients but rather their relative magnitudes.

Second, the coefficients must reflect the actual prices received or paid for the product. If a product is being sold, one should not use prices from distant markets but rather prices adjusted to include marketing costs. Input prices often need to be adjusted to include acquisition costs.

Finally, each objective function coefficient should be developed in harmony with the total model structure. Often, students try to insure that each and every objective function coefficient in a profit maximizing model is the per unit profit contribution arising from that particular variable. This often leads to mistakes and

great confusion. Consider the model

$$\begin{array}{rcl}
 \text{Max} & 3X_1 & \& 2X_2 \\
 \text{s.t.} & X_1 & \& X_2 & ' & 0 \\
 & X_1 & & & \# & 10 \\
 & & & X_2 & \# & 8
 \end{array}$$

Suppose  $X_1$  represents the sale of a commodity and  $X_2$  the purchase of inputs. In order to sell  $X_1$  one must purchase  $X_2$  as reflected by the first constraint. One could use the equality constraint to collapse  $X_1$  and  $X_2$  into a single variable, but this may not be desirable. The contribution of  $X_1$  is fully represented in the above model. The objective function should collectively represent the net margin and one does not need to compute each variable's coefficient so that it is the per unit net contribution.

#### 6.4.6.2 Right hand Side Coefficients

Right hand side coefficients are not always easily specified. For example, consider the amount of labor available. One could think that this is the number of employees times the hours they work a week. However, the nominal and real availability of resources often differs. In the labor context, there are leaves due to sickness, vacation, and alternative assignments diverting labor to other enterprises. Weather can also reduce effective availability. Finally, the right hand sides need to be developed on the same time frame as the rest of the model.

#### 6.4.6.3 Technical Coefficients

The  $a_{ij}$  (technical) coefficients within the model give the resource use per unit of the variables. In developing technical coefficients, one usually uses economic engineering. For example, per unit labor use might be calculated by dividing the total hours of labor by the number of units produced. Such a calculation procedure by its nature includes overhead labor usages such as setup time, cleaning time, etc. However, one needs to be careful in handling fixed usages of labor which do not vary with production.

## 6.5 Purposeful Modeling

The purpose of a modeling exercise influences how a model is implemented. Some variables and constraints become relevant or irrelevant depending upon what exactly is to be done with the model. For example, when studying short-run operating decisions one can omit investment variables and capital constraints. On the other hand, if the focus of the study is investment one may be able to simplify the short-run operating model and come up with an approximation of how investments should be utilized if acquired. Model purpose also has important implications for the specific way a model answer is given to a decision maker.

### 6.5.1 Model Structure

Any problem can be formulated in a number of different ways. Modelers almost always have the option of collapsing items into the objective function or entering them explicitly in the constraints. Often the purpose of a modeling exercise influences model structure (although this is less true when using GAMS than with using conventional methods).

Years ago when LP models were solved by hand or with early LP solvers, it was desirable to construct the smallest possible representation for a particular situation. Today, model condensation is not as desirable because of increased computer and solver capability. Rather, modelers often introduce size increasing features which reduce modeler/analyst interpretation and summarization time. This section discusses ways which study purpose may change formulations (although the discussion is not entirely consistent with our GAMS focus).

Consider a case in which products ( $X_j$ ) are sold at an exogenously fixed price,  $p_j$ . Suppose production utilizes a number of inputs,  $Z_m$ , purchased at an exogenously fixed price,  $r_m$ . Each unit of the production variable ( $Y_k$ ) incurs a direct objective function cost,  $q_k$ , yields ( $a_{jk}$ ) units of the  $j^{\text{th}}$  product and uses  $b_{mk}$  units of the  $m^{\text{th}}$  input. Also, there are constraints on unpriced inputs (i). A formulation is:

$$\begin{aligned}
\text{Max} \quad & \sum_j p_j X_j \quad \& \quad \sum_k q_k Y_k \quad \& \quad \sum_m r_m Z_m \\
\text{s.t.} \quad & X_j \quad \& \quad \sum_k a_{jk} Y_k \quad \leq \quad 0 \quad \text{for all } j \\
& \sum_k b_{mk} Y_k \quad \& \quad Z_m \quad \leq \quad 0 \quad \text{for all } m \\
& \sum_k c_{ik} Y_k \quad \leq \quad e_i \quad \text{for all } i \\
& X_j, \quad Y_k, \quad Z_m \geq 0
\end{aligned}$$

This formulation contains a constraint for each product and each input. One could utilize the first two constraint equations to eliminate  $X_j$  and  $Z_m$  from the model yielding the formulation:

$$\begin{aligned}
\text{Max} \quad & \sum_k g_k Y_k \\
\text{s.t.} \quad & \sum_k c_{ik} Y_k \leq e_i \quad \text{for all } i \\
& Y_k \geq 0
\end{aligned}$$

where the  $g_k$ 's are given by

$$g_k = \sum_j p_j a_{jk} - \sum_m r_m b_{mk} + q_k$$

Suppose the study involves examination of the implications of input and output prices. In the second formulation, these prices are compacted into the objective function coefficients. In the first problem, however, these prices are explicitly included in the objective function. This difference gives a reason why one might prefer the first as opposed to the second formulation. If the prices were to be repeatedly changed, then only one coefficient would have to be changed rather than many. Further, one could easily use cost-ranging features within LP algorithms to study the effects of changes in  $r_m$ . In addition, the solution would report the optimal



production ( $X_j$ ) and input usage ( $Z_m$ ) levels. Post-solution summarization of total yield and input usage would require many calculations under the condensed model, but with  $Z_m$  explicitly included in the formulation, only one number would need to be recorded.

Usage of modeling systems like GAMS places a little different twist on the discussion above as one can easily use GAMS to do post solution report writing and since GAMS computes the whole model every time, changing one or many coefficients makes little difference.

### **6.5.2 Report Writing**

A very important aspect of model use is properly summarizing the output so that understandable information is generated for the decision makers involved with the modeling exercise. This introduces the general topic of report writing.

Linear programming solution reports are generally inadequate for conveying the essence of the solution to the decision maker. It is highly desirable to develop reports which summarize the solution as part of the computer output, possibly an autonomous part. Such reports can be designed to translate the model solution into decision maker language using both the solution results and the input parameters. An example of such report writing is presented in Table 6.10 which gives summary reports on the transportation model from the last chapter. These reports are broken into five tables. The first Table entitled MOVEMENT gives the quantity moving between each pair of cities along with the total movement out of a particular plant and into a particular market. The elements of this table are largely optimal levels of the decision variables in the solution. The second table (COSTS) gives a summary of commodity movements cost by route telling the exact cost of moving between pairs and then the total costs of moving goods out of plants or into markets. This set of outputs is not directly from the linear programming solution, but rather is the cost of movement between a particular city pair times the amount moved. The only number in the table directly from the linear programming output is the objective function value. The third gives a supply use report for each supply point giving the available supply,

the amount shipped out, and the marginal value of that shipment (which is the shadow price). The fourth table gives similar information for the demand markets. Finally, there is the table CMOVEMENT which gives the cost of changing the commodity movement pattern which is a reformat of the reduced costs of the decision variables. In general, the function of a report writer is to summarize the essence of the solution, making it more readable to decision makers. In many applied studies it is valuable to develop a report format ahead of time, then structure the model and model experiments so that the report data are directly generated. The use of computerized report writing instead of hand summaries is a great advantage and can save hours and hours of modeler time. This is particularly facilitated when one uses a computerized modeling system such as GAMS.

## References

- Baker, T. and B.A. McCarl. "Representing Farm Resource Availability over Time in Linear Programs: A Case Study." North Central Journal of Agricultural Economics. 4(1982):59-68.
- Brink, L. and B. McCarl. "The Tradeoff Between Expected Return and Risk Among Cornbelt Farmers." American Journal of Agricultural Economics. 60(1978):259-263.
- Day, R.H. Recursive Programming and Production Response. Amsterdam: North Holland Publishing Co., 1963.
- Heady, E.O. and W.V. Candler. Linear Programming Methods. Ames, Iowa: Iowa State University Press, 1958.
- McCarl, B.A. "GAMSCHECK USER DOCUMENTATION: A System for Examining the Structure and Solution Properties of Linear Programming Problems Solved using GAMS." Working Documentation, Department of Agricultural Economics, Texas A&M University, 1994.
- Sengupta, J. and R. Sfeir. "Allocative Behavior Under Risk Aversion Through Quadratic Programming Experiments." Applied Economics. 12(1980):367-75.

**Table 6.1. Initial Schematic for Example Farm Planning Problem**

	Crop 1	Crop 2	Crop 3	Crop 4	RHS
Objective	$c_1$	$c_2$	$c_3$	$c_4$	
Land	1	1	1		# L
Labor					

**Table 6.2. Revised Schematic for Example Farm Planning Problem**

	Crop 1	Crop 2	Crop 3	Crop 4	RHS
Objective	$c_1$	$c_2$	$c_3$	$c_4$	
Land	1	1	1		# L
Land After Crop 1	-1			1	# 0
Labor					

**Table 6.3. Final Table for Example Farm Planning Problem**

	Crop 1	Crop 2	Crop 3	Crop 4	RHS
Objective	$c_1$	$c_2$	$c_3$	$c_4$	
Land	1	1	1		# L
Land After Crop 1	-1			1	# 0
Labor - Spring	$d_1$	$d_3$	$d_5$		# sp
Labor - Summer	$d_2$			$d_7$	# su
Labor - Fall		$d_4$	$d_6$	$d_8$	# f

**Table 6.4. Composition of a Chicken and Sales Prices for the Component Parts**

	Percent of Chicken Body Weight	Sale Price of Part	Percentage Chicken Meat
	lbs. part/lbs. chicken	\$/lb.	lbs. meat/lb. part
Breast Quarter	50	1.00	75
Leg Quarter	35	.80	60
Neck	10	.20	20
Giblets	5	.70	0

**Table 6.5. Alternative Formulations of Chicken Processing Problem**Formulation 6.5(a)

	Chickens (lbs.)	Breast Quarter (lbs.)	Leg Quarter (lbs.)	Neck (lbs.)	Giblets (lbs.)	Maximize
Objective function (\$)		+ 1.00X <sub>1</sub>	+ 0.80X <sub>2</sub>	+ 0.20X <sub>3</sub>	+ 0.70X <sub>4</sub>	
Balance (lbs.)	-Y	+ 0.50X <sub>1</sub>	+ 0.35X <sub>2</sub>	+ 0.1X <sub>3</sub>	+ 0.05X <sub>4</sub>	# 0
Chickens Available (birds)	1/3Y					# 1500

Formulation 6.5(b)

	Chickens (lbs.)	Breast Quarter (lbs.)	Leg Quarter (lbs.)	Neck (lbs.)	Giblets (lbs.)	Maximize
Objective Function (\$)		+ 1.00X <sub>1</sub>	+ 0.80X <sub>2</sub>	+ 0.20X <sub>3</sub>	+ 0.70X <sub>4</sub>	
Breast Quarter	-0.50Y	+ X <sub>1</sub>				# 0
Leg Quarter	-0.35Y		+ X <sub>2</sub>			# 0
Neck	-0.10Y			+ X <sub>3</sub>		# 0
Giblets	-0.05Y				+ X <sub>4</sub>	# 0
Chickens	1/3Y					# 1500

**Table 6.6. Formulations for Processing Chickens with the Option of Deboning**

Formulation 6.6(a)

	Chicken	Breast Qtr.	Leg Qtr.	Neck	Giblet	Breast Qtr. Meat	Leg Qtr. Meat	Neck Meat	
Objective		$1.0X_1$	$+ 0.8X_2$	$+ 0.2X_3$	$+ 0.7X_4$	$+ 1.2M_1$	$+ 1.2M_2$	$+ 1.2M_3$	
Breast Qtr.	$-0.5Y$	$+ X_1$							# 0
Leg Qtr.	$-0.35Y$		$+ X_2$						# 0
Neck	$-0.1Y$			$+ X_3$					# 0
Giblets	$-0.5Y$				$+ X_4$				# 0
Chickens	$1/3Y$								# 1500
BQ Meat	$-(0.05)(0.75)Y$					$+ M_1$			# 0
LQ Meat	$-(0.35)(0.6)Y$						$+ M_2$		# 0
N Meat	$-(0.2)(0.1)Y$							$+ M_3$	# 0

Formulation 6.6(b)

	Chicken	Breast Qtr.	Leg Qtr.	Neck	Giblet	Breast Qtr. Meat	Leg Qtr. Meat	Neck Meat	Total Meat Sold
Objective		$1.0X_1$	$+0.8X_2$	$+0.2X_3$	$+0.7X_4$				$+ 1.2M_4$
Breast Qtr.	$-0.5Y$	$+ X_1$				$+M_1$			# 0
Leg Qtr.	$-0.35Y$		$+ X_2$				$+M_2$		# 0
Neck	$-0.1Y$			$+ X_3$				$+M_3$	# 0
Giblets	$-0.05Y$				$+ X_4$				# 0
Chickens	$1/3Y$								# 1500
Meat						$- 0.75M_1$	$-0.6M_2$	$-0.2M_3$	$+M_4$

**Table 6.7. Formulations of the Chicken Assembly-Disassembly Problem**

Formulation 6.7(a)											
Chicken											
	BQ	LQ	Neck	Giblet	BQ Meat	LQ Meat	Neck Meat	Total Meat Sold	MQ Sold		
Objective	$1.0X_1$	$+ 0.8X_2$	$+ 0.2X_3$	$+ 0.7X_4$				$+ 1.2M_4$	$+ 0.95Q$		
BQ	$-0.5Y$	$+ X_1$			$+ M_1$				$+ 0.5Q$	$\neq 0$	
LQ	$-0.35Y$	$+ X_2$				$+ M_2$			$+ 0.5Q$	$\neq 0$	
Neck	$-0.1Y$		$+ X_3$				$+ M_3$			$\neq 0$	
Giblets	$-0.05Y$			$+ X_4$						$\neq 0$	
Chickens	$1/3Y$									$\neq 1500$	
Meat					$- 0.75M_1$	$- 0.6M_2$	$- 0.2M_3$	$+ M_4$		$\neq 0$	
Formulation 6.7(b)											
Chicken											
	BQ	LQ	Neck	Giblet	BQ Meat	LQ Meat	Neck Meat	Total Meat Sold	BQ included in MQ	LQ included in MQ	MQ Sold
Objective	$1.0X_1$	$+ 0.8X_2$	$+ 0.2X_3$	$+ 0.7X_4$				$+ 1.2M_4$			$0.95Q$
BQ	$-0.5Y$	$+ X_1$			$+ M_1$				$+ Q_1$		$\neq 0$
LQ	$-0.35Y$	$+ X_2$				$+ M_2$				$+ Q_2$	$\neq 0$
Neck	$-0.1Y$		$+ X_3$				$+ M_3$				$\neq 0$
Giblets	$-0.5Y$			$+ X_4$							$\neq 0$
Chickens	$1/3Y$										$\neq 1500$
Meat					$- 0.75M_1$	$- 0.6M_2$	$- 0.2M_3$	$+ M_4$			$\neq 0$
Qtr. Pack									$- Q_1$	$- Q_2$	$0 Q_3$

**Table 6.8. Alternative LP Formulations of Chair Production Example**

<u>Formulation 6.8(a)</u>									
	Chair Production		Regular Labor		Overtime Labor		Chair Sale		
Objective		-	10	-	15	+	220		
Chairs	-1					+	1	#	0
Regular Labor	7	-	1					#	0
Overtime Labor	3			-	1			#	0
Regular Labor Constraint			1					#	77
Overtime Labor Constraint					1			#	27
<u>Formulation 6.8(b)</u>									
	Chair Production		Regular Labor		Overtime Labor		Chair Sale		
Objective		-	10	-	15	+	220		
Chairs	-1					+	1	#	0
Regular Labor	10	-	1	-	1			#	0
Regular Labor Constraint			1					#	77
Overtime Labor Constraint					1			#	27



**Table 6.9 Example of GAMS Report Writing**

---

----	53	PARAMETER	MOVEMENT	COMMODITY	MOVEMENT	
		MIAMI	HOUSTON	MINEPLIS	PORTLAND	TOTAL
NEWYORK		30	35	15		80
CHICAGO				75		75
LOSANGLS			40		50	90
TOTAL		30	75	90	50	245
----	61	PARAMETER	COSTS	COMMODITY	MOVEMENT	COSTS BY ROUTE
		MIAMI	HOUSTON	MINEPLIS	PORTLAND	TOTAL
NEWYORK		600	1400	525		2525
CHICAGO				1500		1500
LOSANGLS			1400		2000	3400
TOTAL		600	2800	2025	2000	7425
----	68	PARAMETER	SUPPLYREP	SUPPLY	REPORT	
		AVAILABLE	USED	MARG	VALUE	
NEWYORK		100.00	80.00			
CHICAGO		75.00	75.00	15.00		
LOSANGLS		90.00	90.00	5.00		
----	75	PARAMETER	DEMANDREP	DEMAND	REPORT	
		REQUIRED	RECEIVED	MARG	COST	
MIAMI		30.00	30.00	20.00		
HOUSTON		75.00	75.00	40.00		
MINEPLIS		90.00	90.00	35.00		
PORTLAND		50.00	50.00	45.00		
----	80	PARAMETER	CMOVEMENT	COSTS OF	CHANGING	COMMODITY
		MIAMI	HOUSTON	MINEPLIS	PORTLAND	
NEWYORK					75.00	
CHICAGO		45.00	35.00		40.00	
LOSANGLS		75.00		40.00		

---

## CHAPTER VII: MORE LINEAR PROGRAMMING MODELING

In this chapter we continue our concentration on LP modeling. However, we lessen our concentration on GAMS<sup>5</sup> and duality. This presentation is organized around common LP problems. The first problem involves product assembly where component parts are assembled into final products. This is followed by problems which cover: a) raw products disassembled into component parts; b) simultaneous raw product disassembly and component assembly processes; c) optimal operation sequencing; d) commodity storage; e) input output analysis; and f) block diagonal problems.

### 7.1 Assembly Problem

An important LP formulation involves the assembly or blending problem. This problem deals with maximizing profit when assembling final products from component parts. The problem resembles the feed formulation problem where mixed feeds are assembled from raw commodities; however, the assumption of known component mixtures is made. This problem appears in Dano, who presents a brief literature review.

The problem formulation involves k component parts which can be purchased at a fixed price. The decision maker is assumed to maximize the value of the final products assembled less the cost of c components. Each of the final products uses component parts via a known formula. Also, fixed resources constrain the production of final products and the purchase of component parts. The formulation is

$$\begin{array}{ll}
 \text{Max} & \sum_j c_j X_j \quad \& \quad \sum_k d_k Q_k \\
 & \sum_j a_{kj} X_j \quad \& \quad \sum_k w_k Q_k \quad \# \quad h_k \quad \text{for all } k \\
 & \sum_j e_{ij} X_j \quad \% \quad \sum_k f_{ik} Q_k \quad \# \quad b_i \quad \text{for all } i \\
 & X_j \quad \quad \quad \quad \quad \quad \$ \quad g_j \quad \text{for all } j \\
 & X_j, \quad \quad \quad Q_k \quad \$ \quad 0 \quad \text{for all } k, j
 \end{array}$$

where: j is the final product index;  $c_j$  is the return per unit of final product j assembled;  $X_j$  is the number of units of final product j assembled; k is the component part index;  $d_k$  is the cost per unit of component part k;  $Q_k$  is the quantity of component part k purchased;  $a_{kj}$  is the quantity of component part k used in assembling one unit of product j;  $w_k$  is the number of units of the component part received when  $Q_k$  is purchased; I is the index on resource limits;  $e_{ij}$  is the use of limited resource I in assembling one unit of product j;  $f_{ik}$  is the use of the  $i^{\text{th}}$  limited resource when acquiring one unit of  $Q_k$ ;  $b_i$  is the amount of limited resource I available;  $g_j$  is the amount of  $j^{\text{th}}$  product which

---

<sup>5</sup> The GAMS formulations are included on the attached disk in the CH7 subdirectory.

must be sold; and  $h_k$  is the firm's inventory of ingredient  $k$ .

In this formulation the objective function maximizes the return summed over all the final products produced less the cost of the component parts purchased. The first constraint equation is a supply-demand balance and constrains the usage of the component parts to be less than or equal to inventory plus purchases. The second constraint limits the resources used in manufacturing final products and purchasing component parts to the exogenous resource endowment. The last constraint imposes a minimum sales requirement on final product production. All of the variables are assumed to be nonnegative. This problem contains production variables which produce the  $j^{\text{th}}$  product ( $X_j$ ) and purchase variables ( $Q_k$ ).

The dual problem is not very much different from those before, thus, suppose we only look at the dual constraint associated with  $Q_k$ . That constraint

$$w_k U_k \leq \sum_i f_{ik} Z_i + d_k$$

where  $U_k$  is the return to one unit of component part  $k$ ; and  $Z_i$  is the return to one more unit of limited resource  $i$ .

This constraint is more easily interpreted if it is rewritten as follows

$$\sum_i f_{ik} Z_i - d_k \leq w_k U_k$$

or, equivalently,

$$\frac{\sum_i f_{ik} Z_i - d_k}{w_k} \leq U_k$$

This inequality says that the internal value of a component part unit is less than or equal to its purchase price plus the cost of the resources used in its acquisition. Therefore, the internal value of a component part can be greater than the amount paid externally.

### 7.1.1 Example

The assembly problem example involves PC compatible computer assembly by Computer Excess (CE). CE is assumed to assemble one of six different computer types: XT, AT, 80386-25, 80386-33, 80486-SX, and 80486-33. Each different type of computer requires a specific set of component parts. The parts considered are 360K floppy disks, 1.2 Meg floppy disks, 1.44 Meg floppy disks, hard disks, monochrome graphics setups, color graphics setups, plain cases, and fancy cases. The component part requirements to assemble each type of computer are given in Table 7.1. The table also contains component parts' prices, as well as sales, inventory and resource (labor and shelf space) requirements. The resource endowment for labor is 550 hours while there are 240 units of

system space and 590 units of shelf space. The problem formulation is given in Table 7.2 while the solution is given in Table 7.3. The GAMS implementation of this formulation is called ASSEMBLE.

There are no particularly unique features of the empirical formulation or solution, so interpretation is left to the reader.

### 7.1.2 Comments

The assembly problem is related to the feed formulation problem. Namely, the assembly problem assumes that known least cost mixes have been established, and that one wishes to obtain a maximum profit combination of these mixes. There are numerous assumptions in this problem. For example we assume all prices are constant and the quantity of fixed resources is constant. One could extend the model to relax such assumptions.

## 7.2 Disassembly Problems

Another common LP formulation involves raw product disassembly. This problem is common in agricultural processing where animals are purchased, slaughtered and cut into parts (steak, hamburger, etc.) which are sold. The problem is also common in the forest products and petroleum industries, where the trim, cutting stock and cracking problems have arisen. In the disassembly problem, a maximum profit scheme for cutting up raw products is devised. The primal formulation involves the maximization of the component parts revenue less the raw product purchase costs, subject to restrictions that relate the amount of component parts to the amount of raw products disassembled. The basic formulation is

$$\begin{array}{llll}
 \text{Max} & \sum_j c_j X_j & - & \sum_k d_k Q_k \\
 & \sum_j a_{kj} X_j & - & Q_k \leq 0 \quad \text{for all } k \\
 & \sum_j e_{rj} X_j & - & \sum_k f_{rk} Q_k \leq b_r \quad \text{for all } r \\
 & X_j & & \leq g_j \quad \text{for all } j \\
 & & & Q_k \leq h_k \quad \text{for all } k \\
 & & & Q_k \leq M_k \quad \text{for all } k \\
 & X_j, & Q_k & \geq 0 \quad \text{for all } k, j
 \end{array}$$

where  $j$  indexes the raw products disassembled;  $k$  indexes the component parts sold;  $r$  indexes resource availability limits;  $c_j$  is the cost of purchasing one unit of raw product  $j$ ;  $X_j$  is the number of units of raw product  $j$  purchased;  $d_k$  is the selling price of component part  $k$ ;  $Q_k$  is the quantity of component part  $k$  sold;  $a_{kj}$  is the yield of component part  $k$  from raw product  $j$ ;  $e_{rj}$  is the use of resource limit  $r$  when disassembling raw product  $j$ ;  $f_{rk}$  is the amount of

resource limit  $r$  used by the sale of one unit of component part  $k$ ;  $b_r$  is the maximum amount of raw product limit  $r$  available;  $g_j$  is the maximum amount of component part  $I$  available; and  $h_k$  is the maximum quantity of component  $k$  that can be sold, while  $M_k$  is the minimum amount of the component  $k$  that can be sold.

The objective function maximizes operating profit, which is the sum over all final products sold ( $Q_k$ ) of the total revenue earned by sales less the costs of all purchased inputs. The first constraint is a product balance - limiting the quantity sold to be no greater than the quantity supplied when the raw product is disassembled. The next constraint is a resource limitation constraint on raw product disassembly and product sale. This is followed by an upper bound on disassembly as well as upper and lower bounds on sales.

The  $X_j$  are production variables indicating the amount of the  $j^{\text{th}}$  raw product which is disassembled into the component parts (the items produced) while using the inputs  $e_{jr}$ . The  $Q_k$  are sales variables indicating the quantity of the  $k^{\text{th}}$  product which is sold.

### 7.2.1 Example

The disassembly problem example involves operations at Jerimiah's Junk Yard. The firm is assumed to disassemble up to four different types of cars : Escorts, 626's, T-birds, and Caddy's. Each different type of car yields a unique mix of component parts. The parts considered are metal, seats, chrome, doors and junk. The component part yields from each type of car are given in Table 7.4 as are data on car purchase price, weight, disassembly cost, availability, junk yard capacity, labor requirements, component part minimum and maximum sales possibilities, parts space use, labor use, and sales price. The resource endowment for labor is 700 hours while there is 42 units of junk yard capacity and 60 units of parts space. We also extend the basic problem by requiring parts to be transformed to other usages if their maximum sales possibilities have been exceeded. Under such a case, chrome is transformed to metal on a pound per pound basis, while seats become junk on a pound per pound basis, and doors become 70% metal and 30% junk. The problem formulation is given in Table 7.5 while the solution is given in Table 7.6. The GAMS implementation of this formulation is called DISSASSE.

Note the empirical formulation follows the summation notation formulation excepting for the addition of the parts transformation activities and the equality restrictions on the parts balance rows. This is reflected in the solution where, for example, the excess seats are junked making more seats worth the junk disposal cost.

### 7.2.2 Comments

It is difficult to find exact examples of the disassembly problem in literature. This formulation is a rather obvious application of LP which, while having been studied a number of times, is not formally recognized.

A number of observations are possible. First, raw products are assumed to be separable into the individual component parts. While this assumption is not restrictive here, it is more important in the assembly-disassembly problem as discussed below. Second, the product demand curve reflects an finitely inelastic demand at the minimum, then an infinitely elastic portion at the price up until the maximum is met, then zero demand. This is very common in LP, although inventories or minimum requirements would yield slightly different setups. Third, this is an example of the joint product formulation where multiple products are created by the acquisition of the raw products.

### 7.3 Assembly-Disassembly

A unification and extension of the above two models involves the assembly-disassembly problem. In that problem, one purchases raw products, disassembles them and reassembles the component parts into finished products. This type of problem would be most applicable for vertically integrated processing facilities. One example is meat packing, where animals are purchased, disassembled into parts, and then reassembled into such

$$\begin{array}{llllll}
 \text{Max} & \sum_j c_j X_j & - \sum_k d_k Q_k & - \sum_i s_i T_i & + \sum_i p_i Z_i & \\
 & \sum_j a_{ij} X_j & - \sum_k b_{ik} Q_k & - T_i & + Z_i & \leq 0 & \text{for all } i \\
 & \sum_j e_{rj} X_j & - \sum_k f_{rk} Q_k & - \sum_i g_{ri} T_i & + \sum_i h_{ri} Z_i & \leq r & \text{for all } r \\
 & X_j & & & & \leq \bar{X}_j & \text{for all } j \\
 & & Q_k & & & \leq \bar{Q}_k & \text{for all } k \\
 & & & T_i & & \leq \bar{T}_i & \text{for all } i \\
 & & & & Z_i & \leq \bar{Z}_i & \text{for all } i \\
 & X_j, & Q_k, & T_i, & Z_i & \geq 0 & \text{for all } i, j, k
 \end{array}$$

composite products as sausage, ham, processed meats, etc. A similar example would be given by a furniture manufacturer that bought raw logs, cut them up, then used the sawn lumber in a furniture manufacturing business. The basic formulation is

where:  $j$  indexes raw product;  $k$  indexes manufactured product;  $i$  indexes component parts;  $r$  indexes resource limits;  $c_j$  is the cost per unit of raw material;  $X_j$  is the number of units of raw product purchased;  $d_k$  is the return per unit of manufactured product  $k$ ;  $Q_k$  is the number of units of manufactured product  $k$  which are assembled;  $s_i$  is the return the firm realizes from selling one unit of component part  $i$ ;  $T_i$  is the number of units of component parts  $i$  which are sold;  $p_i$  is the per unit purchase price for acquiring component part  $i$ ;  $Z_i$  is the number of units of

component part  $i$  purchased;  $a_{ij}$  is the yield of component part  $i$  from one unit of raw product  $j$ ;  $b_{ik}$  is the use of component part  $i$  to produce one unit of manufactured product  $k$ ;  $e_{ij}$  is the use purchasing one unit of the  $X_j$  raw product makes of the  $r^{\text{th}}$  resource;  $f_{rk}$  is the use manufacturing one unit of  $Q_k$  product makes of the  $r^{\text{th}}$  resource;  $g_{ri}$  is the use that one unit of  $T_i$  makes of the  $r^{\text{th}}$  resource;  $h_{ri}$  is the use one unit of  $Z_i$  makes of the  $r^{\text{th}}$  resource; and  $r$  is the availability of the  $r^{\text{th}}$  resource. In addition each of the four types of variables are bounded above and nonnegative.

This model covers the disassembly of raw products ( $X_j$ ), the assembly of final products ( $Q_k$ ), the sale of component parts ( $T_i$ ), and the purchase of component parts ( $Z_i$ ). The objective function maximizes the revenue from final products and component parts sold less the costs of the raw products and component parts purchased. The first constraint is a supply-demand balance, and balances the use of component parts through their assembly into final products and direct sale, with the supply of component parts from either the disassembly operation or purchases. The remaining equations impose resource limitation constraints and upper bounds. The problem contains production ( $X_j$ ,  $Q_k$ ), sale ( $T_i$ ) and purchase variables ( $Z_i$ ).

### 7.3.1 Example

Charles Chicken Plucking and Sales Company purchases chickens, cuts them and repacks them into chicken meat packages. All chickens are available for \$1.00, have the same weight and breakdown identically in terms of wings, legs, etc. Charles can, however, cut up the chickens in several different manners. Chickens may be cut into parts, meat, quarters, halves, or breast, thigh, and leg cuts. From these Charles gets wings, legs, thighs, backs, breasts, necks, gizzards, meat, breast quarters, leg quarters and halves. Chickens yield 1 lb. of meat, 80 percent of which is in the leg-thigh and breast region. The cutting patterns and yields, labor requirements, and sale prices are shown in Table 7.7.

Charles sells parts individually or sells packs which contain: a) a cut-up chicken with all parts from one chicken except gizzards; b) 4 breast quarters; c) 4 leg quarters; d) 2 chicken halves; and e) two legs and two thighs. Charles also sells gizzard packs with 10 gizzards in them. The individual parts sell at prices shown in Table 7.7.

Production capacity allows 1,000 chickens to be cut up per day. The company may purchase wing, leg, or thighs from other suppliers; however, no more than 20 units of each part are available. The price of purchasing is \$0.02 above the market sales price. The firm has 3,000 units of labor available.

To formulate this problem four classes of variables must be defined: 1) chicken disassembly variables depicting the number of chickens cut via the patterns for - Parts ( $X_p$ ), Halves ( $X_h$ ), Quarters ( $X_q$ ), Meat ( $X_m$ ), and

Leg-Breast-Thigh ( $X_L$ ); 2) assembly variables for the packs A - E ( $X_a - X_e$ ) and the gizzard pack ( $X_g$ ); 3) raw-product sales variables; and 4) raw product purchase variables. These variables are set subject to constraints on part supply, chicken availability, labor and purchase limits. The formulation is shown in Table 7.8. The GAMS implementation is called ASSDISSM.

There are several features of this formulation which merit explanation. First, negative coefficients in the supply and demand rows depict supplies of component parts from either purchase or raw product disassembly, while the positive coefficients depict usages. Examining the thigh row, thighs can be obtained if one cuts a chicken into parts, or into the leg-breast-thigh pattern. Thighs may also be acquired through external purchase. Demand for these thighs comes from the sale of packs of A and E, as well as the direct sale of thighs. Second, the last three constraints give the resource limit constraints on purchases. However, these constraints limit a single variable by placing an upper bound on its value. Constraints which limit the maximum value of a single variable are called upper bound constraints.

The solution to this problem is shown in Table 7.9. This solution leads to an objective function value of \$1362.7 where 1000 chickens are bought, cut by the Leg-Breast Thigh Pattern and sold as 1010 units of pack E. In addition, 2000 breasts, 1000 necks, 200 lbs. meat, 20 thighs and 20 legs purchased. The resultant marginal value of products of the resources are given by the shadow prices.

### **7.3.2 Comments**

There are several important assumptions embodied in the Charles Chicken problem. One of these involves separability -- that the parts can be disassembled and assembled freely. While this assumption appears obvious, let us illustrate an example wherein this assumption is violated. Suppose that one wishes to solve a blending problem mixing two grades of grain (A,B) from two batches ( $G_1, G_2$ ). Suppose that moisture and foreign matter are the component parts, and the relevant parameters are given in Table 7.10. Further, suppose there are 20 units of each of two batches of grain available, and that grade A grain sells for \$3.00 per unit, and grade B grain for \$2.00 per unit.



Now suppose this problem is formulated as an assembly-disassembly problem. This formulation

$$\begin{array}{rllllllll}
 \text{Max} & 3A & \% & 2B & & & & & & & \\
 \text{s.t.} & \& A & \& 2B & \% & 2G_1 & \% & G_2 & \# & 0 \\
 & \& A & \& 2B & \% & G_1 & \% & 2G_2 & \# & 0 \\
 & A & \% & B & \& G_1 & \& G_2 & ' & 0 \\
 & & & & & & G_1 & & & \# & 20 \\
 & & & & & & & & G_2 & \# & 20 \\
 & A, & & B, & & G_1, & & G_2 & \$ & 0 & 
 \end{array}$$

is where A is the quantity of grade A grain made, B the quantity of grade B grain made,  $G_1$  the quantity of batch 1 grain used, and  $G_2$  the quantity of batch 2 grain used (see file GRAIN1). The model balances the maximum moisture, foreign matter, and weight with the amount in each grain. The resultant solution gives an objective function value equal to 100. The variable values and shadow prices are presented in Table 7.11.

There is a problem with this solution. It is impossible, given the data above, to make a mix 20 units each of grade A and grade B grain. The requirement for maximum moisture and foreign matter is 1 percent in grade A grain, so neither of the grain batches could be used to produce grade A, as they both exceed the maximum on this requirement. The solution above, however, implies that we could make 20 units of grade A grain and 20 units of grade B grain. The model uses excess moisture from grain batch 1 in grade B grain, while the excess foreign material from grain batch 2 is also put in the grade B grain. This is clearly impossible, as moisture and foreign matter are not separable (Ladd and Martin make this mistake as pointed out in Westgren and Schrader). Thus, this situation violates the separability assumption; the items in a row cannot be used freely in either of the two blends. The proper formulation of the blending problem is

$$\begin{array}{rllllll}
\text{Max} & 3A & \% & 2B & & & & & \\
\text{s.t.} & \& A & & \% & 2G_{11} & \% & G_{21} & \# & 0 \\
& & \& A & & \% & G_{11} & \% & 2G_{21} & \# & 0 \\
& & & A & & \& G_{11} & \& G_{21} & & ' & 0 \\
& & & & \& 2B & & \% & 2G_{12} & \% & G_{22} & \# & 0 \\
& & & & \& 2B & & \% & G_{12} & \% & 2G_{22} & \# & 0 \\
& & & & & B & & \& G_{12} & \& G_{22} & ' & 0 \\
& & & & & & G_{11} & & \% & G_{12} & & \# & 20 \\
& & & & & & & & & G_{21} & & \% & G_{22} & \# & 20 \\
& & A, & B, & G_{11}, & G_{21}, & G_{12}, & G_{22} & \$ & 0
\end{array}$$

The optimal solution gives an objective function value of 80 (see file GRAIN2). The optimal value of the variables and equation information is shown in Table 7.12.

Here, note that all the grain goes into grade B, and the objective function is smaller. Separability is an important assumption, and one must be careful to insure that it holds in any assembly-disassembly problem.

### 7.4 Sequencing Problems

Often, production entails multiple intermediate processes and requires that each process be completed before the next one is started. Furthermore, the intermediate processes often compete for resources. A farming example of this situation involves the requirement that plowing be done before planting and that planting be done before harvest. However, plowing and planting may go on at the same time on different tracts of land. Thus, plowing and planting could draw from the same labor and machinery pools. Similarly, plowing could be done in the fall during the harvesting period, thus harvesting and fall plowing compete for the same resources.

A problem which explicitly handles such a situation involves sequencing. Sequencing models insure that the predecessor processes are completed before the successor processes can begin. In LP models the sequencing considerations generally take one of two forms. Sequencing may be controlled within a variable or between variables using constraints. Sequencing within a variable is done whenever the occurrence of one event implies that another event occurs a fixed time afterward, or whenever the timing of events influences their economic returns. In this case both the predecessor and successor tasks are embedded in a variable. Sequencing between variables is done whenever the successor process follows the predecessor process an indefinite amount of time later, but the

economic return to the successor is not a function of when the predecessor was done. For example, one may plant a crop and not care when it was plowed; on the other hand, if one harvests a crop and the yield of the crop depends on both the planting and harvesting dates, then this would require sequencing within a variable. Both cases will be illustrated.

Sequencing considerations are hard to write algebraically and the result is often confusing to students. Thus, we will alter our presentation style and use an example before the algebraic formulation. Suppose a firm produces an output using two tasks. Further suppose that production process occurs somewhere in a three week period, and that the successor or predecessor can be done in any of those three weeks. However, the predecessor task must be completed before the successor task. Two cases can arise. First, the successor/predecessor date could jointly determine economic returns and/or resource use. Second, the yield, returns, etc., could be independent of timing as long as the predecessor occurs first.

Let us consider the latter case first. Suppose we denote  $X_1$ ,  $X_2$  and  $X_3$  as the amount of the predecessor done in weeks 1, 2, and 3, and  $Y_1$ ,  $Y_2$  and  $Y_3$  as the amount of the successor done in weeks 1, 2, and 3. In week 1 the level of the successor activity ( $Y_1$ ) must be less than or equal to the amount of the predecessor activity ( $X_1$ ) completed. Algebraically, this implies

$$Y_1 \leq X_1 \quad \text{or} \quad X_1 \geq Y_1 \geq 0$$

In week 2, the amount of the predecessor activity which could be completed by then is  $X_1+X_2$ . The total amount of the successor activity at that time would be that used this week ( $Y_2$ ), plus that used last week  $Y_1$ . The sequencing requirement is that  $Y_2$  must be less than or equal to  $X_1$  plus  $X_2$  minus what was used in Period 1 or, algebraically

$$Y_2 \leq X_1 + X_2 - Y_1 \quad \text{or} \quad X_1 + X_2 - Y_1 \geq Y_2 \geq 0$$

The intersection of this constraint with that above allows no more of the successor activity to be nonzero in the first week than is present at that time. However, more of the predecessor may be produced than used in the first week; with the extra carried over into later weeks for usage.

In the third week, the amount of the successor activity ( $Y_3$ ) is less than or equal to the amount of predecessor activity that could be supplied up to that period ( $X_1 + X_2 + X_3$ ) less that used in periods 1 and 2 ( $Y_1 + Y_2$ ).

$$Y_3 \leq X_1 + X_2 + X_3 - Y_1 - Y_2 \quad \text{or} \quad X_1 + X_2 + X_3 - Y_1 - Y_2 \geq Y_3 \geq 0$$

Algebraically, the action of this set of constraints is such that  $X_1$  could equal 500, while  $Y_1=100$ ,  $Y_2=100$

and  $Y_3=300$ , indicating that the predecessor activity was completely undertaken in the first week but the successor activity slowly used up the inventory during the life of the model. However, the successor cannot get ahead of the predecessor. A complete tableau of this setup is given by

Week1	& $X_1$			%	$Y_1$		#	0
Week2	& $X_1$	& $X_2$		%	$Y_1$	%	$Y_2$	# 0
Week3	& $X_1$	& $X_2$	& $X_3$	%	$Y_1$	%	$Y_2$	% $Y_3$ # 0
Week1	$aX_1$			%	$dY_1$			# $T_1$
Week2		$bX_2$				%	$eY_2$	# $T_2$
Week3			$cX_3$				%	$fY_3$ # $T_3$

The first three constraints are discussed above. The others are resource limitations. Week 1 resources are used by  $X_1$  and/or  $Y_1$ ; Week 2 resources by  $X_2$  and/or  $Y_2$ ; and Week 3 by  $X_3$  and/or  $Y_3$ . The sequencing constraints insure that successor activities will not be undertaken until the predecessors are complete.

The sequencing restraints may also be explained in terms of the variables. Suppose  $X_1$  and  $Y_1$  are variables for acres of land.  $X_1$  supplies land for use in time periods 1, 2 and/or 3.  $Y_1$  requires land for use in period 1 and precludes use in periods 2 and 3. Variable  $X_2$  supplies land in periods 2 and 3 while  $Y_2$  uses land in period 2 and precludes use in period 3.

The above formulations assume that returns and resource usage are independent of activity timing. This may not always be true. Returns to the successor activities may depend on the timing of the preceding activities. Such a formulation involves changing the variable definitions so that the week of the predecessor and successor define the variable. In the above example, this yields six variables - the first involving the successor and predecessor both carried out in week 1; the second, the predecessor in week 1 and the successor in week 2, etc. A tableau of this situation is

Predecessor date	Wk 1	Wk 1	Wk 1	Wk 2	Wk 2	Wk 3		
Successor date	Wk 1	Wk 2	Wk 3	Wk 2	Wk 3	Wk 3		
Wk 1	$aZ_{11}$	+	$bZ_{12}$	+	$dZ_{13}$			# $T_1$
Wk 2		$cZ_{12}$		+	$fZ_{22}$	+	$gZ_{23}$	# $T_2$
Wk 3			$eZ_{13}$		+	$hZ_{23}$	+	$iZ_{33}$ # $T_3$

The variables represent the amount of predecessor and successor activities undertaken during specific times (ie.  $Z_{13}$  involves the predecessor in week 1 and the successor in week 3). The constraints restrict resource usage by week. Resource use or objective function coefficients would differ from activity to activity, indicating that features of the model are dependent on the sequencing.

Given this background, we may now introduce a general formulation. Suppose we have three phases, X, Y and Z, each of which must be completed in sequence. Further, we will allow a set of alternatives for each of X, Y, and Z. A general summation model embodying sequencing considerations is

$$\begin{array}{ll}
 \text{Max} & \sum_j c_j X_{jt_1} \quad \sum_k d_k Y_{kt_2} \quad \sum_s e_s Z_{st_3} \\
 \text{s.t.} & \sum_{j: t_1 \leq t} X_{jt_1} \quad \sum_{k: t_2 \leq t} Y_{kt_2} \quad \sum_{s: t_3 < t} Z_{st_3} \quad \# \quad 0 \quad \text{for } t = t_1, t_2 \\
 & \sum_j a_j X_{jt} \quad \sum_k b_k Y_{kt} \quad \sum_s f_s Z_{st} \quad \# \quad g_{mt} \quad \text{for all } m, t \\
 & X_{jt}, \quad Y_{kt}, \quad Z_{st} \quad \geq \quad 0 \quad \text{for all } j, k, s, t_1, t_2, t_3
 \end{array}$$

The variables are  $X_{jt_1}$ , the  $j^{\text{th}}$  alternative for the completion of task X in time period  $t_1$ ;  $Y_{kt_2}$ , the  $k^{\text{th}}$  alternative for the completion of task Y in time period  $t_2$ ; and  $Z_{st_3}$  the  $s^{\text{th}}$  alternative for the completion of task Z in time period  $t_3$ . The first two constraints depict sequencing as in the first example where the predecessor activities are summed as long as they precede the period over which the constraint is defined (denoted by  $t_1 \leq t$ ). These constraints are defined for the time periods in which the successor activities begin. The third constraint depicts resource availability.

A formal definition of notation is:  $t$  designates time periods;  $j$  indexes the technologies by which the first task can be done;  $k$  indexes the technologies by which the second task can be done;  $s$  indexes the technologies by which the third task can be done;  $t_1$  gives the periods in which the first task can be done;  $t_2$  gives the periods of the second task;  $t_3$  gives the periods of the third task;  $c_j$  is the cost of a unit of first task  $j$ ;  $X_{jt_1}$  is the number of units of first task  $j$  performed in time period  $t_1$ ;  $d_k$  is the cost of a unit of second task  $k$ ;  $Y_{kt_2}$  is the number of units of second task  $k$  performed in period  $t_2$ ;  $e_s$  is the revenue of a unit of third task version  $s$ ;  $Z_{st_3}$  is the number of units of third task version  $s$  performed in period  $t_3$ ;  $a_{jm}$  is the number of units of resource  $m$  used by the  $j^{\text{th}}$  alternative for X;  $b_{km}$  is the number of units of resource  $m$  used by the  $k^{\text{th}}$  alternative for Y,  $f_{sm}$  is the number of units of resource  $m$  used by the  $s^{\text{th}}$  alternative for Z;  $g_{mt}$  is the endowment of resource  $m$  in period  $t$ .

There are no particularly new features to this formulation in terms of types of constraints and/or variable. All the variables are some form or another of a production variable. The first two constraints are supply/demand balances on the intermediate products passed between the predecessor and successor

variables, and the last constraint is a resource limitation constraint.

Due to the complexity of the above formulation, two examples will be given, one straight forward but of limited realism, the other more complex.

#### 7.4.1 Example 1

Suppose that a farmer plows, discs, plants, cultivates, and harvests land and that yield does not depend on activity timing. Assume plowing is done in April, May and June. The principal resources are land and plowing labor. Plowing labor usage is 0.2 hours of labor per acre with a cost of \$100 per acre. Discing follows plowing. Suppose that discing is done in May, June and July, requiring 0.3 hours of labor per acre while costing \$20 per acre. Planting is done in May, June and July, requiring 0.3 hours of labor per acre and costing \$25 per acre. Cultivation is done in each of the three months following planting, and harvesting must be done in the fourth month. Cultivation requires 0.1 hours of labor per acre with no added cost, while harvesting uses 0.5 hours of labor per acre plus a cost of \$75 per acre. Further, suppose that the crop yield is worth \$500 an acre. Also, the farm's resource endowment is 600 acres of land and 160 hours of labor in each of the months of April through November.

A LP formulation of this situation is given in Table 7.13 and the GAMS implementation of this formulation is called SEQUEN. The activities X represent the plowing possibilities and are defined for April, May and June. The activities Y represent the discing possibilities and are defined for May, June and July. The activities Z represent the planting-harvesting possibilities and are defined by beginning month - May, June, and July. The second three constraints give the link between plowing and discing. Note these constraints are defined for the periods in which discing can be started - May, June and July. The next three constraints are the link between discing and planting and are again defined for May, June and July. The next eight constraints are labor constraints which are defined for each of the months during which farm activity can be done. The last constraint is the land constraint.

The planting variable Z encompasses the sequencing between planting, cultivating and harvesting since those activities are done in fixed sequence. This variable's coefficients require resources committed in May be accompanied by a commitment in June, July and August for cultivating and in September for harvesting. The objective function for these activities is calculated as margin between the cost of planting and harvesting and the value of the crop sold. Thus, the return is \$400 per acre. The solution to this model is given in Table 7.14.

This solution depicts 600 acres of plowing in April followed by 407.41 acres of discing in May and 192.59 acres in July. Finally, 125.93 acres are planted in May, 281.48 in June and 192.59 in July. The sequencing constraints allow the predecessor to occur initially faster than they are used by the successor, although the successor eventually catches up. Labor has no shadow prices. Land is the only binding resource constraint.

#### **7.4.2 Example 2**

The second example is more complex and does not closely follow the summation notation. However, it does contain the sequencing considerations reflected above. This example again reflects a farm planning situation and illustrates what needs to be done when planting and harvesting date influence yield. Assume that a farm grows two crops. The crops are plowed in March through June. Plowing is done the same way regardless of the crop, and the plowing rate is four acres per hour. In addition, the farmer needs one hour of maintenance for each 20 acres of plowing, and the plowing cost is \$5.00 an acre. Both crops are then disced. Discing can be done in April-June for crop 1 and March-June for crop 2. The farmer can disc five acres an hour of either crop. Crop 2 is always disced exactly one month preceding planting. Discing of crop 1 can be done any time before planting. Discing costs \$3.00 per acre.

Both crops are planted in April-June. The farmer figures it takes 0.22 hours of labor to do one acre. The planting cost for crop 1 is \$40 per acre, and the cost for crop 2 is \$20 per acre. Both crops are cultivated exactly one month after planting. The farmer can cultivate 10 acres in one hour, the cost for cultivation is \$10 per acre.

The yield achieved depends on the crop planting and harvest dates and is given in Table 7.15. Harvesting takes 0.7 hours per acre for crop 1 and 0.6 hours for crop 2. Harvesting costs are \$10 per acre.

The farm has 1500 acres and 300 hours of labor available in each month. Labor can be hired for \$10 an hour. The objective of the model is to maximize profits. Crop 1 sells for \$3.00 per unit, Crop 2 for \$8.70 per unit.

A formulation of this problem is given in Table 7.16. The GAMS implementation of this formulation is called FARM2. The tableau is formed with plowing variables defined in March through June. These variables use land and supply plowed land for the subsequent discing activities. The discing operation is modeled in two different ways, depending on crop. For crop 1 there are explicit discing variables; for crop 2, the discing activities are embedded into the planting activity because of the one month

sequencing requirement. Crop 1 planting variables are defined for April, May and June. We also add the harvest time to the variable definition. Cultivating resource usages are also included. Thus, the variable for a crop planted in April and harvested in September includes resource use for April planting, May cultivating, and September harvest. Plowing resource usage is computed as  $1/(\text{acres per hour}) + 1/(\text{maintenance time per acre}) = 0.25 + 0.05$  or 0.3 hours for the total. Similarly, the discing time is  $1/(\text{acres per hour}) = 0.20$

The crop 2 planting and harvesting activities reflect a slightly different setup. Here, discing is also included. Thus, the resources use accounts for discing, planting, cultivation, and harvest. Also note that the yields are entered in the bottom of the tableau and are sold through the selling activities.

The resultant solution to this problem is given in Table 7.17 and indicates that \$449,570 is made from the farm with 775 acres planted to crop 1 and remainder planted to crop 2. Labor is hired only in October. The value of land is \$292.50 per acre; labor is worth \$10 an hour in March, April, and October, which means the labor is fully exhausted up until the point at which it will need to be hired. Labor is in slack in June and July and worth \$3 in May and September.

### 7.4.3 Comments

The above sequencing problem is a special case of a much larger class of sequencing problems. The particular problem we present has the predecessor followed by only one type of successor task. There have been many differently setup sequencing problems. For example the PERT/CPM project scheduling problems allow multiple following activities to occur (see Bradley, Hax and Magnanti). Job shop scheduling problems are related (Bradley, Hax and Magnanti). Most of the scheduling problems, however, are integer programming problems. In addition, numerous sequencing problems have been formulated and solved which involve sequencing between years. Such problems are examined in the dynamics chapter.

The second example illustrates an important general point. That is, in applied modeling, we often should not reflect annual resource constraints (i.e. labor for the year) but, rather, period by period resource constraints within a year. Often resources in some periods of the year are not perfect substitutes with resources in other periods of the year (Heady and Candler). Farming provides an obvious example where labor during the winter is not substitutable with labor during the harvest or planting periods. Similarly, labor during the planting periods is not usually substitutable with labor during the harvest period.



## 7.5 Storage Problems

Problems can involve storage where a product resource or input can be retained between time periods. LP has been used to analyze such problems virtually since its inception, (see Dantzig [1963] and Gass (1985) for a historical perspective). This section reviews a general formulation of a relatively simple storage problem. Ordinarily, a storage problem would not be solved alone but, rather would be a model subcomponent.

Assume a decision maker is planning over a time horizon involving  $T$  periods and has a single item in inventory which can be stored or sold in period. Let us assume that the decision maker incurs a storage cost in carrying the item from one time period to the next. The decision problem involves maximizing the value of the sales less storage costs subject to storage capacity. Thus, the problem is to determine the optimum sale and holding policy. We will also include constraints on maximum storage capacity and the maximum/minimum amount which can be sold in any time period. Further, it will be assumed that the inventory is not increased at any time during the time horizon. Thus, the storage constraint is only active in the first time period. The formulation of the problem is

$$\begin{array}{ll}
 \text{Max} & \sum_{t=1}^T c_t X_t \quad \& \quad \sum_{t=1}^{T-1} cs_t H_t \\
 \text{s.t.} & X_1 \quad \leq \quad H_1 \quad \# \quad s_0 \\
 & X_t \quad \& \quad H_{t+1} \quad \leq \quad H_t \quad \# \quad 0 \quad \text{for all } 1 < t < T \\
 & X_T \quad \& \quad H_{T+1} \quad \leq \quad 0 \\
 & X_t \quad \leq \quad U_t \quad \text{for all } t \\
 & X_t \quad \geq \quad L_t \quad \text{for all } t \\
 & H_1 \quad \leq \quad S \\
 & X_t, H_t \quad \geq \quad 0 \quad \text{for all } t
 \end{array}$$

$X_t$  stands for the amount sold in the  $t^{\text{th}}$  period;  $H_t$  stands for the amount held over from time period  $t$  to time period  $t+1$ ;  $c_t$  stands for returns from sales of the item in time period  $t$ ;  $cs_t$  stands for the cost of storing from period  $t$  into period  $t+1$ ;  $s_0$  the amount of inventory available in the first time period;  $U_t$  is the upper limit on the sales possibility in time  $t$ ;  $L_t$  is the lower limit on the sales possibility in time period  $t$ ;  $S$  is the total storage capacity limit which only is binding on the amount stored in the first time period during which the greatest amount would be stored.

The first equation in the model is the objective function. It involves summation across all the periods of the revenues from the sales of the good less the costs of storage of the good. We only include storage from the time periods 1 through T-1, assuming that everything must be sold in the last time period. The first constraint limits the quantity sold in the first period plus the quantity stored into the second period to be less than or equal to the initial inventory available. The next constraints are active in all time periods excepting 1 and T. This limits the amount sold in each period plus the amount stored into the next period to not exceed the amount held over from the period before. The third constraint gives the inventory condition for the last time period requiring that sales not exceed inventory carried over from the time period before. This constraint precludes outgoing storage because we are assuming everything has to be sold by the last period. Similarly, the first constraint does not include incoming storage except as an exogenous quantity on the right hand side. The next two constraints impose upper and lower limits on the amount that can be sold during any time period. The last constraint imposes an upper limit on storage in the first period. Additional constraints on storage capacity are not needed for the subsequent periods as stored amount cannot increase.

This problem does not contain new types of constraints and variables. However, it does use a new form of the transformation variable. The H variables transform the time utility of an item by moving it from one time period to another at a cost. The X variables are again sales activities. The constraints are a mixture of resource limitations and supply demand balances.

Insight can be gained into the model solution properties considering the dual. The dual constraint associated with the variable  $X_t$  states that the value of the item in time period t must be greater than or equal to the revenue from selling it less the costs of the upper bound plus the costs of the lower bound. Assuming there are no bounds, this constraint then would require that the value of grain be no less than the price at which it could be sold. The dual constraints associated with the storage activities insure a relationship between the marginal values of grain in adjacent periods where they state that the value of grain in time period t+1 must be less than or equal to its value in time period t plus the cost of storing it between the periods. This occurs at optimality since, if the sale price of grain in time period t+1 is greater than or equal to the sale price of grain in time t by more than the storage costs, then it would be economical to store. The only exception comes in relation to period 1, where storage is limited. If storage is binding there may be a larger wedge between the first two shadow prices than the cost of storage.

### 7.5.1 Example

Suppose a farmer has 100 bushels of a crop available to sell over four time periods. The farmer expects the price in the first time period to be \$2.30; in the second, \$2.50; in the third, \$2.70; and in the fourth, \$2.90. The farmer expects the cost of holding grain from time period 1 to time period 2 to be \$.10 a bushel; from time period 2 to time period 3, \$.20 a bushel; and from time period 3 to time period 4, \$.30 a bushel. The farmer cannot sell any more than 50 bushels in any one time period. Also, the farmer must (for cash flow reasons) sell at least 15 bushels in the first time period and at least 5 in the second. Finally, the farmer has no more than 75 bushels of storage capacity available. A formulation of this is presented in Table 7.18. The GAMS implementation of this formulation is called STORE.

The solution to this problem is given in Table 7.19. The solution has 25 bushels of grain sold in time period 1; these are sold because there is not enough capacity to store into the second time period as reflected by the \$.10 shadow price on the overall storage constraint. Then, 50 bushels of grain are sold in the second time period limited by the upper limit on the ability to sell in the time period. Subsequently, 25 bushels are sold in the third time period. The inventory pattern shows 75 bushels carried from the first to the second period with 25 carried from the second to the third. The first four shadow prices show the marginal values of grain in each time period. The fifth equation exhibits slack of 25 bushels indicating 25 more bushels could be sold in the first time period.

### 7.5.2 Comments

Several assumptions could be relaxed in this formulation. One could allow inventory to be replenished. Storage costs could be made a function of volume and/or one could allow acquisition of storage capacity. The model is commonly used in conjunction with a planning problem to develop an overall aggregate plan (as discussed in Holt et al.), wherein production and storage decisions are jointly determined. More examples of storage and dynamic carry-over will be presented in the dynamic modeling chapter.

Another comment involves the nature of the dual relationships that a transformation variable imposes. Transformation variables bound the maximum difference between the shadow prices. This is an absolute property in any LP solution containing such variables. This is exhibited in our solution where the storage variables cause the shadow prices to differ by no more than the storage cost.

## 7.6 Input Output Analysis

Another application of linear programming involves the use of Leontiefs' (1951) input output (IO) formulation. However, IO formulations are not used alone in an LP context as there is specialized software for IO models. Rather IO's are used in conjunction within larger LP models (i.e. see Penn et al.).

Leontief formulated the IO model and it has been utilized in a number of different contexts since then (Miernyk). The IO model fundamentally deals with the development of the economy wide implications of changes in the export level and/or production practices within the economy.

The fundamental data in setting up an IO model involves three things. First, there is an identification of the sectors. These sectors are divided into endogenous and exogenous sectors. The endogenous sectors are integral parts of the economy which purchase and sell items to other sectors within the economy. The exogenous sectors are those from which imports are made or those to which goods and services flow or export. Given identification of these, one develops the transaction matrix which tells how the endogenous sectors within the economy dispose of the revenue that they receive for production. These endogenous sectors completely dispose of the revenue that they receive across the endogenous and exogenous sectors. Thus, the model data accounts for the full distribution of earnings including that to exogenous sectors and retained earnings. The third data item is the final demand vector. This tells the value of goods that flow outside the local economy.

Suppose we identify four components of an economy. There are endogenous sectors: 1) manufacturing, 2) services, and 3) agriculture with an exogenous sector for other uses of funds which includes imports and retained earnings. The transaction matrix gives the distribution of revenue by sector. For example, in the manufacturing sector, that matrix would tell how much was spent: a) within the manufacturing sector on manufactured goods from other producers; b) on the service sector for purchasing services used in the manufacturing process; c) on agricultural goods; and d) on acquisitions from or distributions to exogenous parties. In turn, this would also be done for the service and households sectors. The household sector data gives the household consumption function showing how the gross household income is distributed across purchases of manufactured goods, services and other household activities as well as flows into the exogenous category which accounts for savings and purchases from outside the economy. Finally, final demand gives the money flow from outside the economy into the manufacturing, service, and agricultural sectors.

After formation of the transactions matrix one then turns to forming the technical coefficients matrix. This tells the proportion of revenue spent by sector. The parameters of the technical coefficient

matrix  $(a_{ij})$  given the transaction matrix  $t_{ij}$  equal:

$$a_{ij} = \frac{t_{ij}}{\sum_k t_{kj}}$$

The denominator term is total revenue to the  $j^{\text{th}}$  sector. All the coefficients will be less than or equal to one indicating the proportion spent within each sector.

In turn then, given this technical coefficients matrix one can form the fundamental equation of input output analysis which states that

$$X = Y + AX$$

where  $Y$  is final demand,  $A$  is the technical coefficients matrix, and  $X$  is a vector giving the amount of total activity in the economy. This indicates that the total economic activity is equal to the amount of final demand plus the value of the intermediate products that are necessary in order to produce that final demand. In turn, this equation can be manipulated to say

$$X - AX = Y$$

which indicates the total production minus the intermediate use needed to support that total production equals exports or

$$(I - A)X = Y.$$

Now  $I - A$  in this case is a square matrix with one positive entry for each of the sectors of the economy.

Providing  $I - A$  is invertible then the solution is

$$X = (I - A)^{-1}Y$$

This is the computational procedure in conventional Input Output analysis. However, in a linear programming context one utilizes the fundamental equation and an objective function to obtain input output solutions. We use the objective function recommended by Brink and McCarl (1977) which maximizes the sums of the activity across all sectors. The resulting formulation is

$$\begin{aligned} \text{Max} \quad & \sum_j X_j \\ \text{s.t.} \quad & \sum_j (I_{ij} - A_{ij})X_j = Y_i \text{ for all } i \\ & X_j \geq 0 \text{ for all } j \end{aligned}$$

where  $X$ ,  $A$  and  $Y$  are as defined before and  $I_{ij}$  is an identity matrix (possessing entries of 1 if  $I = j$  and zero otherwise). This particular formulation should lead to all the  $X$ 's being nonzero with the dual variables equaling the total amount of activity induced in the sector from a change in the right hand side (final demand). This duality property can be argued as follows: if all the  $X$ 's are nonzero then the LP basis

inverse should equal  $(I - A)^{-1}$  and since all the objective function coefficients of the basic variables are ones, then the dual variables will equal

$$\sum_i (I - A)_{ij}^{-1}$$

or should equal the total induced activity inside the model by one unit of additional exports the classic output multiplier (Miernyk).

### 7.6.1 Example

Consider an example with four sectors: manufacturing, agriculture, finance, households and imports. Suppose that the relevant transaction matrix is given in Table 7.20 as is the final demand vector. Under these circumstances the technical coefficient matrix is Table 7.21 and one sets up the linear programming model with the first three sectors as endogenous as in Table 7.22. The GAMS implementation of this formulation is called INOUT. The solution is given in Table 7.23. This solution shows that sectoral activity is \$250 for manufacturing, \$122 for agriculture, \$75 for finance and \$230 for services, but that this activity is used to supply a lesser amount of final demand. Namely, there is \$250 of production by manufacturing to directly deliver \$75 of final demand. Across all sectors, \$677 of total production is needed to satisfy the \$145 of final demand. Thus, there is \$532 worth of intermediate production. Also, notice that the values of the shadow prices on the manufacturing balance row is 4.615. This implies that in order to meet one dollar worth of demand, \$ 4.615 worth of total manufactured products need to be created.

### 7.6.2 Comments

Input-Output models have been used in many contexts (See Miernyk for a review).

## 7.7 Block Diagonal

Many of the early LP problems involved a "block diagonal" structure (see Dantzig (1963) for a historical perspective). Models depicting production in several different locations and/or time periods exhibited such a structure. The name arose since the models contained blocks of constraints and activities which did not overlap with other sets of constraints and other activities. The blocks arise when individual production units utilize immobile resources. The problem also depicts some usage of unifying resources at the overall firm level. The primal formulation of the block diagonal problem is given by

$$\text{Max} \quad \sum_k c_k X_k + \sum_{jL} d_{jL} Y_{jL}$$

$$\begin{aligned}
 \text{s.t.} \quad & \sum_k a_{ik} X_k + \sum_{j,L} g_{ijL} Y_{jL} \leq b_i && \text{for all } I \\
 & \sum_j e_{jLM} Y_{jL} \leq f_{LM} && \text{for all } L \text{ and } M \\
 & X_k, Y_{jL} \geq 0 && \text{for all } k, j \text{ and } L
 \end{aligned}$$

where:  $k$  indexes variables for the overall firm;  $j$  indexes variables for the separate entities;  $L$  indexes entities;  $I$  indexes overall resources;  $M$  indexes separate entity resources;  $c_k$  is the per unit return of overall firm variable  $k$ ;  $X_k$  is the number of units produced of overall firm variable  $k$ ;  $d_{jL}$  is the per unit return of separate entity  $L$ 's  $j^{\text{th}}$  production variable;  $Y_{jL}$  is the number of units produced of separate entity  $L$ 's  $j^{\text{th}}$  production variable;  $a_{ik}$  is the use of overall firm resource  $I$  by  $X_k$ ;  $g_{ijL}$  is the use of overall firm resource  $I$  by  $Y_{jL}$ ;  $b_i$  is the endowment of overall firm resource  $I$ ;  $e_{jLM}$  is the use of the  $m^{\text{th}}$  resource at location  $L$  by  $Y_{jL}$ ; and  $f_{LM}$  is the endowment of resource  $M$  at location  $L$ .

The decision variables  $X_k$  reflect actions at the unifying level of the problem, while the  $Y_{jL}$  reflect actions at the sublevels of the problem. Generally, the sublevels arise because of spatial, temporal or functional separations. The first constraint is the overall unifying set of constraints. The second set of constraints deal with each sub-unit. The problem maximizes profit summed over the global and sub-unit activities subject to an overall linking constraint and individual sub-unit constraints. Note that the second set of constraints do not involve sums across  $L$ , thus only one sub-unit is involved in any constraint. Thus, the second set of constraints is independent across the sub-units. This particular problem is called block diagonal because of the structure of the second constraint. An overview of this problem is given below

$$\begin{array}{rcccccccc}
 cX & \% & d_1 Y_1 & \% & d_2 Y_2 & \cdot & \cdot & \cdot & \% & d_n Y_n & \text{max} \\
 AX & \% & g_1 Y_1 & \% & g_2 Y_2 & \cdot & \cdot & \cdot & \% & g_n Y_n & \# & b \\
 & & e_1 Y_1 & & & & & & & & \# & f_1 \\
 & & & & e_2 Y_2 & & & & & & \# & f_2 \\
 & & & & & & & & & & & \cdot \\
 & & & & & & & & & & & \cdot \\
 & & & & & & & & & & & \cdot \\
 & & & & & & & & & & & e_n Y_n \# f_n
 \end{array}$$

All the cells in this matrix which do not have entries or dots are 0's and the name "block diagonal" comes from the diagonal blocks representing each of the subcomponent resource constraints and activities. This formulation also illustrates the phenomena of sparsity. Linear programming often possesses large blocks of zero coefficients. This particular one is famous for it. Sparsity is commonly exploited in

LP algorithms.

### 7.7.1 Example

The block diagonal problem will be illustrated using the data from the resource allocation example above, adding features regarding multiple plants and the making of tables. Suppose the firm from the resource allocation example has expanded and now possesses 3 plants. At these plants they fabricate and sell chairs and tables which are sold individually or together as dinette sets. Plant 1 makes only tables, plant 2 only makes chairs, and plant 3 can make chairs and tables. The costs and input usages for the chairs and tables are the same across all three plants, and the chair data is in the above example. Making functional tables requires three hours of labor and one unit of top capacity. It takes five hours of labor to make fancy tables and one unit of top capacity. Functional tables involve direct costs of \$80 per unit and fancy tables \$100 per unit. Chairs can be sold either at their point of production or can be transported to an overall assembly point which is located at plant 1. Chairs cost \$5 to transport to the assembly point from plant 2 and \$7 from plant 3. Tables cost \$20 to ship in from plant 3 to the assembly point. Functional tables sell for \$200 when sold alone and fancy tables sell for \$300. The firm can sell functional sets containing four chairs and one table for \$600 and fancy sets containing six chairs and one table for \$1100. Resource endowments in plant 1 are 175 units of labor and 50 units of top capacity. Plant 2 has 140 units of small lathe capacity, 90 units of large lathe capacity, 120 units of chair bottom capacity, and 125 units of labor. Plant 3 has 130 units of small lathe capacity, 100 units of large lathe capacity, 110 units of chair bottom capacity, 210 units of labor, and 40 units of top capacity.

This problem's objective is to maximize net returns to the total firm operation subject to the overall firm linking considerations and the various constraints arising at the plants. The resultant formulation is in Table 7.24. The GAMS implementation is called BLOCKDIA. Note the block diagonal structure involving the resources at plants 1, 2, and 3. Also note the linking considerations on tables and chairs.

The solution to this problem is given in Table 7.25. The variables  $Y_{10}$  and  $Y_{11}$  are the table fabrication activities for plant 1. Variables  $Y_{20} - Y_{29}$  are the transportation and fabrication activities for plant 2, and the variables  $Y_{30} - Y_{35}$  are the transportation and fabrication activities for plant 3. The objective function value is \$36,206.9. The firm manufactures tables and chairs in plants across the various locations as the reader may determine by investigating the solution.

### 7.7.2 Comments

This type of formulation appears in many places in the literature, usually linking other



formulations. In the example, we have used elements of the transportation, resource allocation, and assembly problems. Generally, many LP problems look like this where multiple formulations are combined together in the analysis of a particular problem. Other classes of block diagonal structures appear in the integer programming and dynamic model sections. Dynamic models can be made block diagonal by incorporating the transfer activities from one period to the next into the overall structure. However, dynamic models generally contain a second type of block angular structure where there is some overlap; for example, production in period  $t$  is carried into period  $t+1$  using storage.

The final comment involves the solution to this problem. Dantzig and Wolfe observed that one could solve the block diagonal problem as a set of independent problems providing one had an estimate of the shadow prices on the overall linking constraints. This led to the Dantzig-Wolfe decomposition algorithm which formally exploits problem structure.

### **7.8 Concluding Comments**

A number of LP formulations have been presented in this and the preceding chapter. These have all been simplified and are not easily applicable to any actual case. However, we hope the material increases the reader's familiarity with common usages of LP and shows how formulations may be combined in the analysis of empirical problems. There are a number of other additional comments which arise out of the above.

First, we hope the reader gained appreciation for empirical modeling issues. Data are never directly available rather, they must be calculated. We illustrate this in our examples where, for example, the coefficients required calculation through economic engineering or deductive accounting. Readers should also gain an appreciation for sparsity, as many LP problems contain few non-zero coefficients. This is exploited in the modern implementations of the simplex method where only nonzeros are stored, and fancy re-inversion schemes are utilized to solve the problems exploiting matrix sparsity (Orchard and Hays; Murtaugh).

In addition, we hope the reader has developed an appreciation for the implications of the particular primal variable structures on the shadow prices as arise through duality. Generally, the dual restraints require that the marginal cost of any variable be greater than or equal to the marginal revenue arising under that variable. Specific forms of this for any problem can be discovered by examining the dual. Often, such an exercise clears up confusion regarding shadow price values.

Another important point involves the role of summation notation. In many of the examples we

were able to generate summation notation representations which later exactly translated into empirical models. This provides an important way of thinking about the problem and getting its structure right. Subsequently, one can easily check the properness of the structure and the properties of the dual variables. We believe this is important for modelers, as they can utilize summation representations to generate small example problems which are typical of larger structures. This allows one to try out structures in small empirical problems and also provides guidelines from which computer implementations may be written. It also provides an important way of thinking about and analyzing the overall structure of the problem without concentrating a great deal on the particular empirical numbers involved.

Finally, we would like to mention that the coverage above is by no means comprehensive. We will present many LP problems in the subsequent chapters. There are also LP problems such as the input/output problem (Dorfman, Samuelson and Solow), the trim problem (Eisemann and Golden), and the caterer problem (Jacobs), along with many others which will not be included in this book. The reader should see: a) the literature cited in Riley and Gass, Day and Sparling, and Assad and Golden; b) the presentations and literature cited in such texts as Hillier and Lieberman; Gass (1985); Wagner (1969), Williams (1985); Bradley, Hax, and Magnanti; or Salkin and Saha, along with many others; and c) the many articles appearing in such journals as Management Science, Operations Research, Decision Sciences, Mathematical Programming, American Journal of Agricultural Economics, Canadian Journal of Agricultural Economics, Western Journal of Agricultural Economics, North Central Journal of Agricultural Economics, and Southern Journal of Agricultural Economics.

## References

- Assad, A. and B. Golden. "A Categorized Bibliography of Survey Articles in Management Science and Operations Research." Management Science. 4(1982):425-438.
- Bradley, S.P., A.C. Hax and T.L. Magnanti. Applied Mathematical Programming. Reading, MA: Addison-Wesley Publishing Company, 1977.
- Brink, L. and Bruce McCarl. "Input-Output Analysis, Linear Programming and the Output Multiplier." Canadian Journal of Agricultural Economics. 25(1977):62-67.
- Dan, S. Linear Programming in Industry: Theory and Applications. New York: Springer-Verlag Inc., 1974
- Dantzig, G.B. Linear Programming and Extensions. Princeton, New Jersey, Princeton University Press. 1963.
- Dantzig, G.B. and P. Wolfe. "The Decomposition Algorithm for Linear Programming." Econometrica. 29(1961):767-778.
- Day, R.H. and E. Sparling. "Optimization Models in Agricultural and Resource Economics." in G.G. Judge, R.H. Day, S.R. Johnson, G.C. Rausser and L.R. Martin (eds.). A Survey of Agricultural Economics Literature. Vol 2, 1977.
- Dorfman, R., P.A. Samuelson, and Robert M. Solow. Linear Programming and Economic Analysis. New York: McGraw-Hill Book Company, Inc. 1958.
- Eismann, K. "The Trim Problem." Management Science 3(1957):279-284.
- Gass, S.I. Linear Programming: Methods and Applications. New York: McGraw-Hill Book Company, Inc. 1958.
- Golden, B. "Approaches to the Cutting Stock Problem." American Institute of Industrial Engineers Transactions. 8(1976):265-274.
- Greenberg, M.R. Applied Linear Programming For the Socioeconomic and Environmental Sciences. New York: Academic Press, 1978.
- Heady, E.O. and W.V. Candler. Linear Programming Methods. Ames, IA: The Iowa State University Press, 1958.
- Hiller, F.S. and G.J. Lieberman. Introduction to Operations Research. San Francisco, CA: Holden Day, 1974.
- Holt, C.C., F. Modigliani, J.F. Muth, and H.A. Simon. Planning Production Inventories and Work Force. Englewood Cliffs, NJ: Prentice-Hall, 1960.
- Jacobs, W.W. "The Caterer Problem." Naval Research Logistics Quarterly. 1(1954):154-165.
- Ladd, G.W. and M.B. Martin. "Prices and Demands for Input Characteristics." American Journal of Agricultural Economics. 58(1976):21-30.
- Leontief, W.W. The Structure of American Economy 1919-1939. 2nd ed., Fair Lawn, NJ: Oxford University Press, 1951.
- McCarl, B.A., W.V. Candler, D.H. Doster, and P.R. Robbins. "Experiences with Farmer Oriented Linear Programming for Crop Planning." Canadian Journal of Agricultural Economics. 25(1977):17-30.
- McCarl, B.A. and P.L. Nuthall. "Linear Programming for Repeated Use in the Analysis of Agricultural Systems." Agricultural Systems. 8(1982):17-39.

- Miernyk, W.H. The Elements of Input-Output Analysis. New York: Random House Inc., 1965.
- Orchard Hays, W. Advanced Linear Programming Computing Techniques. New York: McGraw-Hill, 1968.
- Penn, J.B., B.A. McCarl, L.G. Brink, and G. Irwin. "Modeling and Simulation of the U.S. Economy with Alternative Energy Availabilities." American Journal of Agricultural Economics. 58(1976):663-671.
- Prevatt, J.W., B.E. Melton, T.H. Spreen, and W.K. Mathis. "The Economics of Carcass Beef Production: An Appraisal of Florida's Feedlot Potential." Southern Journal of Agricultural Economics. 10(1978):49-55.
- Riley, V. and S.I. Gass. Linear Programming and Associated Techniques. Published for Operations Research Office. Baltimore, MD: The Johns Hopkins Press, 1958.
- Salkin, H.M. and J. Saha. Studies in Linear Programming. New York: American Elsevier, 1975.
- Wagner, H.M. Principles of Operations Research With Application to Managerial Decisions. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1969.
- Westgren, R.E. and L.F. Schrader. "Prices and Demands for Input Characteristics: Comment." American Journal of Agricultural Economics. 59(1977):592-593.
- Williams, H.P. Model Building in Mathematical Programming. 2nd Edition. Chichester; New York: John Wiley and Sons, 1985.

**Table 7.1. Data for Computer Excess Example****Components Required to Assemble a System**

	XT	AT	386SX	38633	486SX	48633
360FLOPPY	1	1				
12MFLOPPY	1	1	2	1	1	1
144MFLOPPY				1	1	1
HARDDISK		1	1	1	1	1
MONO	1	1	1			
COLORVGA				1	1	1
PLAINCASE	1	1	1			
FANCYCASE				1	1	1

**Components Parts Acquisition Information**

Name	Cost	Inventory	Labor	Shelf Space
360KFLOPPY	35	20	0.01	0.01
12MFLOPPY	49	29	0.01	0.01
144MFLOPPY	52	32	0.01	0.01
HARDDISK	245	45	0.03	0.03
MONO	102	15	0.07	1.50
COLORVGA	302	45	0.10	2.00
PLAINCASE	41	11	0.15	1.70
FANCYCASE	80	12	0.12	1.70

**Final Products Assembly and Sales Information**

Name	Sales Price	Minimum Sales	Assembly Cost	Labor	Space
XT	689	1	59	2.00	1
AT	992	3	102	2.05	1
386SX	1200	2	100	2.21	1
38633	1400	4	300	2.24	1
486SX	1500	2	400	2.18	1
48633	1800	2	700	2.12	1

**Table 7.2. Tableau of Computer Excess Example**

	Assembly										Buy							RHS
	XT	AT	386SX	38633	486SX	48633	360k	12M	144M	HARD	MONO	CVGA	PLAIN	FANCY				
OBJECTIVE	630	890	1100	1100	1100	1100	-35	-49	-52	-245	-102	-302	-41	-80	Max			
360KFLOPPY	1	1				-1									# 20			
12MFLOPPY	1	1	2	1	1	1	-1								# 29			
144MFLOPPY				1	1	1		-1							# 32			
HARDDISK	1	1	1	1	1	1				-1					# 45			
MONO	1	1	1							-1					# 15			
COLORVGA				1	1	1					-1				# 45			
PLAINCASE	1	1	1										-1		# 11			
FANCYCASE				1	1	1								-1	# 12			
LABOR	2	2.05	2.21	2.24	2.18	2.12	0.01	0.01	0.01	0.03	0.07	0.1	0.15	0.12	# 550			
SHELFSPACE							0.01	0.01	0.01	0.03	1.5	2.0	1.7	1.7	# 590			
SYSTEMSPC	1	1	1	1	1	1									# 240			
Lower Bound	1	3	2	4	2	2												

**Table 7.3. Solution for Computer Excess Example**

Objective 155330.097					
Variable	Value	Reduced Cost	Constraint	Slack	Shadow Price
XT	1.0	-168.4	360KFLOPPY	16	0
AT	3.0	-159.1	12MFLOPPY	0	50.9
386SX	172.9	0.0	144MFLOPPY	0	53.9
38633	41.0	0.0	HARDDISK	0	250.7
486SX	2.0	0.0	MONO	0	385.4
48633	2.0	0.0	COLORVGA	0	343.4
360KFLOPPY	0.000	-36.9	PLAINCASE	0	362.2
12MFLOPPY	365.772	0.0	FANCYCASE	0	401.2
144MFLOPPY	13.000	0.0	LABOR	10.09	0
HARDDISK	175.886	0.0	SHELFSPACE	0	188.9
MONO	161.886	0.0	SYSTEMSPC	18.11	0
COLORVGA	0.000	-336.5			
PLAINCASE	165.886	0.0			
FANCYCASE	33.0	0.0			

**Table 7.4. Data for Jerimiah Junk Yard Example**

Car Data	ESCORTS	626S	TBIRDS	CADDIES
PURCHASE PRICE	85	90	115	140
WEIGHT	2300	2200	3200	3900
DISASSEMBLY COST	100	120	150	170
AVAILABILITY	13	12	20	10

Resource Use to Breakdown Cars				
	ESCORTS	626S	TBIRDS	CADDIES
CAPACITY	1	1	1.2	1.4
LABOR	10	12	15	20

Proportional Breakdown of Cars into Component Parts				
	ESCORTS	626S	TBIRDS	CADDIES
METAL	.60	.55	.60	.62
SEATS	.10	.10	.06	.04
CHROME	.05	.05	.09	.14
DOORS	.08	.10	.10	.07
JUNK	.17	.20	.15	.13

Part Data	MINIMUM	MAXIMUM	PRICE	PARTSPACE	LABOR
METAL	0		0.15	0	0.0010
SEATS	4000	6000	1.00	0.003	0.0015
CHROME	70		0.70	0.0014	0.0020
DOORS	2	5000	0.70	0.0016	0.0025
JUNK			-0.05	0	0.0001



**Table 7.5. Tableau of Jerimiah Junk Yard Example**

	ESCORTS	626S	TBIRDS	CADDIES	METAL	SEATS	CHROME	DOORS	JUNK	CONVERT SEATS	CONVERT CHROME	CONVERT DOORS	RHS MIN
OBJECTIVE	-185	-210	-265	-310	0.15	0.90	0.70	1.00	-0.05				= 0
METAL	-1380	-1210	-1920	-2418	1						-1	-0.7	= 0
SEATS	-230	-220	-192	-156		1				1			= 0
CHROME	-115	-110	-288	-546			1				1		= 0
DOORS	-184	-220	-320	-273				1				1	= 0
JUNK	-391	-440	-480	-507					1	-1		-0.3	= 0
CAPACITY	1	1	1.2	1.4									# 42
LABOR	10	12	15	20	.001	.0015	.0020	.0025	.0001				# 700
PARTSPACE						.003	.0014	.0016					# 60
LOWER BOUND						4000	70	2					
UPPER BOUND	13	12	20	10		6000	10000	5000					

**Table 7.6. Solution for Jerimiah Junk Yard Example**

			Objective = 18337.2		
Variable	Value	Reduced Cost	Constraint	Slack	Shadow Price
ESCORTS	4.00	0	Parts		
626S	0	-49.960	METAL	0	0.150
TBIRDS	20	31.688	SEATS	0	-0.050
CADDIES	10	91.356	CHROME	0	0.150
			DOORS	0	0.090
Sell			JUNK	0	-0.050
METAL	73186.2	0	CAPACITY	0	24.760
SEATS	6000	0.95	LABOR	43.512	0
CHROME	10000	0.550	PARTSPACE	20	0
DOORS	5000	0.910			
JUNK	18013.8	0			
Convert					
SEATS	320	0			
CHROME	1680	0			
DOORS	4866	0			

**Table 7.7. Data for Chicken Example Yields from Cutting**

	Parts	Halves	Quarters	Meat	Leg-Breast-Thigh
Wings	2				
Legs	2				2
Thighs	2				2
Back	1				
Breasts	1				2
Necks	1				1
Gizzards	1	1	1	1	
Meat		0.05	0.07	1	0.2
Breast Quarter			2		
Leg Quarter			2		
Halves		2			

**Selling Price and Labor Use for Chicken Packs**

Pack	Labor	Price
A	2	\$2.05
B	1.3	2.00
C	1.2	1.45
D	1.1	1.95
E	1.25	1.25
Gizzard	1.0	0.90

**Individual Selling Prices for Parts**

Part	Price	Part	Price
Wings	0.10	Gizzards	0.07
Legs	0.20	Meat	2.00/lb.
Thighs	0.25	Breast Quarters	0.45
Backs	0.12	Leg Quarter	0.40
Breasts	0.33	Halves	0.90
Necks	0.05		

Table 7.8. Primal Formulation of Charles Chicken Company Problem

	Disassemble										Assemble										Sell										Buy										RHS	
	$X_p$	$X_h$	$X_g$	$X_m$	$X_L$	$X_s$	$X_b$	$X_c$	$X_d$	$X_e$	$X_g$	$X_a$	$X_b$	$X_c$	$X_d$	$X_e$	$X_f$	$X_g$	$X_h$	$X_i$	$X_j$	$X_k$	$X_l$	$X_m$	$X_n$	$X_o$	$X_p$	$X_q$	$X_r$	$X_s$	$X_t$	$X_u$	$X_v$	$X_w$	$X_x$	$X_y$	$X_z$	Max				
Object	-1	-1	-1	-1	-1	2.05	2.00	1.45	1.95	1.25	.90	.10	.20	.25	.12	.33	.05	.07	2.0	.45	.40	.90	-.12	-.22	-.27																	
Wings	-2					2						1																													#	
Legs	-2				-2	2				2			1																													#
Thighs	-2				-2	2				2				1																												#
Backs	-1					1																																				#
Breasts	-1				-2	1																																				#
Necks	-1				-1	1																																				#
Gizzards	-1				-1						10																															#
Meat					-2																																					#
Breast Qtr.					-2																																					#
Leg Qtr.					-2																																					#
Halves					-2																																					#
Chickens	1	1	1	1	1																																					#
Labor						2	1.3	1.2	1.1	1.25	1																															#
Wing																																										#
Leg																																										#
Thigh																																										#

**Table 7.9. Solution to the Charles Chicken Co. Problem**

Objective function = 1362.7					
Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
X <sub>p</sub>	0	-0.22	Wings	0	0.120
X <sub>h</sub>	0	0	Legs	0	0.355
X <sub>q</sub>	0	-0.33	Thighs	0	0.270
X <sub>m</sub>	0	-0.27	Backs	0	0.180
X <sub>L</sub>	1000	0	Breasts	0	0.330
X <sub>a</sub>	0	0	Necks	0	0.050
X <sub>b</sub>	0	0	Gizzards	0	0.090
X <sub>c</sub>	0	-0.15	Meat	0	2.000
X <sub>d</sub>	0	-0.22	Breast Qtr.	0	0.500
X <sub>e</sub>	1010	0	Leg Qtr.	0	0.400
Gizzards	0	0	Halves	0	1.085
Wings	0	-0.02	Chickens	0	1.36
Legs	0	-0.02	Labor	1737.5	0
Thighs	0	-0.155			
Backs	0	-0.06			
Breasts	2000	0			
Necks	1000	0			
Gizzards	0	-0.02			
Meat	200	0			
Breast Qtr.	0	-0.05			
Leg Qtr.	0	0			
Halves	0	-0.185			
Wings	0	0			
Legs	20	0			
Thighs	20	0.135			

**Table 7.10. Data for the Grain Blending Example**

	Grade		Characteristics	
	Maximums		Grain Batch 1	Grain Batch 2
	A	B		
Moisture	1	2	2	1
Foreign Matter	1	2	1	2

**Table 7.11. Solution of the First Formulation of the Grain Blending Problem**

Objective = 100					
Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
A	20	0	Moisture	0	1
B	20	0	Foreign Matter	0	0
G <sub>1</sub>	20	2	Weight	0	4
G <sub>2</sub>	20	3			

**Table 7.12. Optimal Solution to the Correct Formulation of the Grain Blending Problem**

Objective = 80					
Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
A	0	0	1	0	1
B	40	0	2	0	1
G <sub>11</sub>	0	0	3	0	5
G <sub>12</sub>	20	0	4	20	0
G <sub>21</sub>	0	0	5	20	0
G <sub>22</sub>	20	0	6	0	2
			7	0	2
			8	0	2

<b>Table 7.13. LP Formulation of Sequencing Example 1</b>												
		Plow - X			Disc - Y			Plant etc. - Z			RHS	
		April	May	June	May	June	July	May	June	July		
Obj		-100	-100	-100	-20	-20	-20	400	400	400	max	
X - Y link	May	-1	-1		1						#	0
	June	-1	-1	-1	1	1					#	0
	July	-1	-1	-1	1	1	1				#	0
Y - Z link	May				-1			1			#	0
	June				-1	-1		1	1		#	0
	July				-1	-1	-1	1	1	1	#	0
Labor	April	0.2									#	160
	May		0.2		0.3			0.3			#	160
	June			0.2		0.3		0.1	0.3		#	160
	July						0.3	0.1	0.1	0.3	#	160
	Aug.							0.1	0.1	0.1	#	160
	Sept.							0.5	0.1	0.1	#	160
	Oct.								0.5	0.1	#	160
	Nov.									0.5	#	160
Land		1	1	1							#	600



**Table 7.14. Solution to Sequencing Example 1**

Objective function = 168,000							
Variable		Value	Reduced Cost	Equation		Slack	Shadow Price
Plow	April	600	0	Plow-Disc	May	-192.59	0
	May	0	0 (alt)		June	200.00	0
	June	0	0 (alt)		July	0	380
Disc	May	407.41	0	Disc-Plant	May	88.89	0
	June	0	0		June	0	0
	July	192.59	0		July	0	400
Plant	May	125.93	0	Labor	April	97.78	0
	June	281.48	0		May	0	0
	July	192.59	0		June	0	0
			July		0	0	
			Aug.		100	0	
			Sept.		11.11	0	
			Oct.		51.11	0	
			Nov.		60	0	
			Land		0	280	

**Table 7.15. Yields for Crops 1 and 2 by Crop Planting and Harvest Dates**

Harvest Date	Planting Date					
	Crop 1			Crop 2		
	April	May	June	April	May	June
September	110	105	90	38	40	35
October	125	120	118	35	38	40

Table 7.16. Formulation of Problem for Sequencing Example 2

Rows	Plow in Month						Disk for Crop 1 in Month						Plant and Harvest Crop 1 in Months						Disk Plant and Harvest Crop 2 in Months						Hire Labor						Sell		RHS						
	Mar		Apr		May		Jun		Apr		May		Jun		Sep		Oct		Oct		Oct		Mar		Apr		May		Jun		Sep			Oct		Nov		Crop 1	Crop 2
	Mar	Apr	May	Jun	Mar	Apr	May	Jun	Sep	Oct	Sep	Oct	Sep	Oct	Sep	Oct	Sep	Oct	Mar	Apr	May	Jun	Sep	Oct	Nov														
Objective	-5	-5	-5	-5	-5	-5	-3	-3	-3	-3	-3	-60	-60	-60	-60	-60	-60	-60	-43	-43	-43	-43	-43	-43	-43	-10	-10	-10	-10	-10	-10	-10	3	8.7	Max				
Land Balance	1	1	1	1	1	1																													f 1500				
Plowed	-1																																		f 0				
Land	-1	-1																																	f 0				
Balance	-1	-1	-1	-1	-1	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		f 0				
Disced																																				f 0			
Land																																				f 0			
Balance																																				f 0			
Labor	0.3																																			f 300			
Avail-		0.3																																		f 300			
ability			0.3																																	f 300			
Yield																																					f 300		
Crop 1																																				f 0			
Crop 2																																					f 0		

**Table 7.17. Solution for Sequencing Example 2**

Objective function = 449,570							
Variable		Value	Reduced Cost	Equation	Slack	Shadow Price	
Acreage Plowed in:	March	1275	0	Land	0	292.5	
	April	0	0	Plowed Land:	March	1275	
	May	225	0		April	0	2.10
	June	0	0		May	0	14.4
	Acreage Discd for Crop 1 in:	April	775		0	June	0
May		0	0		Discd Land:	April	0
June		0	0	May		0	5.34
Acreage of Crop 1 planted/harvested in:	Sept./April	0	-40.15	June		0	287.0
	Sept./May	0	-49.81	Labor:	March	0	10
	Sept./June	0	-92.65		April	0	10
	Oct./April	775	0		May	0	3
	Oct./May	0	-9.66		June	200.5	0
	Oct./June	0	-13.5		July	277.5	0
	Acreage of Crop 2 planted/harvested in:	Sept./April	0		-19.24	Sept.	0
Sept./May		500	0		Oct.	0	10
Sept./June		0	-39.34	Yield:	Crop 1	0	3
Oct./April		0	-49.5		Crop 2	0	8.7
Oct./May		0	-21.56				
Oct./June		225	0				
Labor hired in:		March	82.5		0		
	April	125.5	0				
	May	0	-7				
	June	0	-10				
	July	0	-10				
	Sept.	0	-6.93				
	Oct.	377.5	0				
Crop 1 Sales		96875	0				
Crop 2 Sales		29000	0				

**Table 7.18. Formulation of Storage Example**

Objective	Sell	Store
	$2.3X_1 + 2.5X_2 + 2.7X_3 + 2.9X_4$	$-.1h_1 - .2h_2 - .3h_3$

Grain Inven- t-ory	1	$X_1$			+	$h_1$				#	100	
	2		$X_2$		-	$h_1$	+	$h_2$		#		
	3			$X_3$			-	$h_2$	+	$h_3$	#	0
	4				$X_4$				-	$h_3$	#	0
Max Sales	1	$X_1$								#	50	
	2		$X_2$							#	50	
	3			$X_3$						#	50	
	4				$X_4$					#	50	
Min Sales	1	$X_1$								\$	15	
	2		$X_2$							\$	5	
Max Store						$h_1$				#	75	

**Table 7.19. Primal Solution to the Storage Problem Example**

Objective = 237.5

Variable	Value	Reduced Cost	Constraint	Slack	Shadow Price
$X_1$	25	0	Pd1 Inventory	0	2.3
$X_2$	50	0	Pd2 Inventory	0	2.5
$X_3$	25	0	Pd3 Inventory	0	2.7
$X_4$	0	0	Pd4 Inventory	0	2.9
$h_1$	75	0	Max sale Pd1	25	0
$h_2$	25	0	Max sale Pd2	0	0
$h_3$	0	-0.1	Max sale Pd3	25	0
			Max sale Pd4	50	0
			Capacity	0	0.1
			Min sale Pd1	10	0
			Min sale Pd2	45	0
			Min sale Pd3	25	0
			Min sale Pd4	0	0

**Table 7.20. Input Output Example Data**

	Transactions Matrix			
	Manufacturing	Agriculture	Finance	Services
Manufacturing	50	40	10	75
Agriculture	20	10	2	40
Finance	25	8	12	20
Services	100	40	40	40
Exogenous	55	24	11	55

**Final Demand Data**

Sector	Final Demand for Sectors
Manufacturing	75
Agriculture	50
Finance	10
Services	10

**Table 7.21. Technical Coefficient Matrix for Input Output**

	Manufacturing	Agriculture	Finance	Services
Manufacturing	0.200	0.328	0.133	0.326
Agriculture	0.080	0.082	0.027	0.174
Finance	0.100	0.066	0.160	0.087
Services	0.400	0.328	0.533	0.174
Exogenous	0.220	0.197	0.147	0.239

**Table 7.22. LP Formulation of Input Output Example**

	Manufacturing	Agriculture	Finance	Services		
Maximize	1	1	1	1		
Manufacturing	0.8	-0.33	-0.13	-0.33	#	75
Agriculture	-0.08	0.92	-0.03	-0.17	#	50
Finance	-0.1	-0.07	0.84	-0.09	#	10
Services	-0.4	-0.33	-0.53	0.83	#	10

**Table 7.23. Solution for Input Output Example**

Objective = 677					
Variable	Value	Reduced Cost	Constraint	Slack	Shadow Price
Manufacturing	250	0	Manufacturing	0	4.615
Agriculture	122	0	Agriculture	0	4.716
Finance	75	0	Finance	0	4.960
Services	230	0	Services	0	4.547

Table 7.24. Matrix Formulation of Block Diagonal Problem

	PLANT 1					PLANT 2					PLANT 3					RHS																								
	Sell Sts FC FY	Make Table FC FY	Sell Table FC FY	Transport Chair FC FY	Sell Chair FC FY	Make Functional Chairs Norm MxSm MxLg	Make Fancy Chairs Norm MxSm MxLg	Transport Table FC FY	Transport Chair FC FY	Sell Table FC FY	Sell Chair FC FY	Make Table FC FY	Make Functional Chairs Norm MxSm MxLg	Make Fancy Chairs Norm MxSm MxLg																										
Objective	600	1100	-80	-100	200	300	-5	-5	82	105	-15	-16	-15.7	-25	-26.5	-26.6	-20	-7	200	300	82	105	-80	-100	-15	-16	-15.7	-25	-26.5	-26.5	Max									
P L A	1	FC	-1		1												-1															f	0							
N T	4	FC	-1				-1										-1																f	0						
	6	Inventory					-1																											f	0					
I		Labor	3	5																															f	17				
		Top Capacity	1	1																															f	5				
P L A N T		Chair	FC				1		1																											f	0			
		Inventory					1																														f	0		
		Small Lathe											0.8	1.3	0.2																					f	14			
		Large Lathe											0.5	0.2	1.3																					f	0			
		Chair Bottom										0.4	0.4	0.4																							f	90		
		Carver																																			f	12		
		Labor											1	1.05	1.1																					f	0			
																																					f	12		
P L A N T		Table	FC																																			f	5	
		Inventory																																				f	0	
		Chair	FC																																				f	0
		Inventory																																					f	0
		Small Lathe																																					f	13
		Large Lathe																																					f	0
																																							f	10
																																							f	0

Chair Bottom																					
Carver																					
Labor																					3 5
Top Capacity																					1 1

**Table 7.25. Primal Solution to the Block Diagonal Problem**

Objective = 36206.9

Variable		Value	Reduced Cost	Equation	Slack	Shadow Price	
Plant1	Sell FC set	24.40	0	Plant1	FC Tables	0 212	
	Sell FY set	29.01	0		FY Tables	0 320	
	Make FC Table	24.40	0		FC Chairs	0 97	
	Make FY Table	20.36	0		FY Chairs	0 130	
	Sell FC Table	0	-12		Labor	0 44	
	Sell FY Table	0	-20		Top Cap	5.240 0	
	Trans FC Chair	62.23	0		Plant2	FC Chair	0 92
Plant2	Trans FY Chair	78.2	0	Plant2	FY Chair	0 125	
	Sell FC Chair	0	-10		Sm Lathe	0 47.77	
	Sell FY Chair	0	-20		Lrg Lathe	0 38.83	
	Make FC Table	0	-58.11		Chair Bot	16.907 0	
	Make FY Table	0	-96.85		Labor	0 19.37	
	Make FC Chair N	62.23	0		Plant3	FC Table	0 200
	Make FC Chair MS	0	-14.2			FY Table	0 300
	Make FC Chair ML	0	-5.04			FC Chair	0 90
	Make FY Chair N	73.02	0			FY Chair	0 123
	Make FY Chair MS	0	-10.24			Sm Lathe	0 18.50
	Make FY Chair ML	5.18	0			Lrg Lathe	0 12.19
Plant3	Trans FC Table	0	-8	Plant3	Chair Bot	0 35.27	
	Trans FY Table	8.649	0		Labor	0 40.00	
	Trans FC Chair	35.37	0		Top Cap	20.562 0	
	Trans FY Chair	95.85	0				
	Sell FC Table	0	0				
	Sell FY Table	10.79	0				
	Sell FC Chair	0	-8				
	Sell FY Chair	0	-18				
	Make FC Table	0	0				
	Make FY Table	19.44	0				
	Make FC Chair N	35.37	0				
	Make FC Chair MS	0	-8.59				



Make FC Chair ML	0	-3.35
Make FY Chair N	76.83	0
Make FY Chair MS	0	-6.68
Make FY Chair ML	19.02	0

## CHAPTER VIII: MULTI-YEAR DYNAMICS AND LINEAR PROGRAMMING

Many problems contain multiple year dynamic elements (hereafter called dynamics). This is especially true in agricultural problems covering perennial crops, livestock and/or facility investments. Dynamic concerns arise when decision makers face: a) binding financial constraints which change with time (i.e. cash flow considerations); b) a production situation in which current actions impact the productivity of future actions (e.g., crops in rotation, livestock breeding or equipment purchases); c) a need to adjust over time to exogenous stimulus (e.g., altering production as prices or resource endowments change); d) future uncertainty and/or e) an exhaustible resource base. This section presents modeling techniques for incorporating dynamic considerations into linear programs. The discussion will be limited to the modeling of multiple-period situations. Within-period (year) dynamics are covered in the sequencing example of the previous chapter. Before beginning a discussion of methods, we first present background principles.

### 8.1 Dynamics Background

A number of key questions are involved with the modeling of dynamic situations. Fundamentally, one must ask whether an explicit multiple time period representation is necessary. If so, a number of other questions are relevant. First, the length of the total time period and the starting date must be determined. Second, the length of the time intervals explicitly represented within the total time period must be determined. Third, initial and final inventory conditions must be specified. Fourth, one must decide on activity life; i.e., when a particular activity is begun and how long it lasts. Fifth, the rate of time preference must be determined; i.e., one needs the discount rate at which future returns are considered when compared with current returns. Sixth and finally, one must decide whether to include uncertainty. The sections below present discussion on each of these topics.

#### 8.1.1 Should Dynamics be Explicit?

Dynamic situations may not require multi-period dynamic models. Some dynamic questions must be explicitly modeled, allowing the solution to change over time. On the other hand, other questions may be adequately depicted by a steady state equilibrium model. In an equilibrium model the same decision is assumed to be repeatedly made in all time periods and thus a "representative" single period representation is used.

Choice between these two modeling alternatives depends on a number of considerations. First, one must ask whether modeling adaptation is important. This depends upon whether the modeled entity is likely to experience growth, development/exhaustion of its resource base, and/or dynamic changes in model parameters. Second, one must be interested in the time path of adjustment and must not be content to solve a model for an optimal final state with the adjustments required to attain that state determined exogenously. Simultaneously, one must ask whether the data are present in sufficient detail to support a dynamic model. Finally, the multi-period dynamic analysis must be affordable or practical given the model size and data required.

Dynamic equilibrium models may be used when one is willing to assume: a) the resource, technology and price data are constant; and b) a long-run "steady state" solution is acceptable. Disequilibrium models are used when these assumptions do not hold. Often reliance on equilibrium models is stimulated by the absence of data on parameter values over time.

The decision on whether or not to assume equilibrium needs to be addressed carefully. Two common errors occur in the context of dynamic models: unnecessarily entering explicit dynamics into a model and improperly omitting them. Naturally, the proper dynamic assumption depends upon the problem. Treating dynamics as an equilibrium does not imply ignoring dynamics, but rather assumes repetitive decision making with equal initial and final inventory, a zero growth rate and a constant resource base.

### **8.1.2 How Long?**

Given that a dynamic model is to be used, one must determine the length of time that the model covers. There are trade-offs between time coverage and size. Longer lengths generally add variables and constraints along with data needs.

Determination of the model time frame involves many considerations. One of the simplest, yet powerful statement comes from Madigliani and Cohen, who stated that the time horizon must be long enough so that alterations in its duration do not impact the initial period solution. However, this statement is based on the

assumption that the first period variables are the main focus of interest. When more periods are of interest (e.g., first five years), an obvious restatement is that the time frame should be long enough so that its extension does not alter the variables of interest. Such a criteria, while appealing, may not be terribly practical. Model size may limit its attainment. An alternative modeling strategy is to specify a number of periods explicitly, then introduce terminal conditions for in-process inventories. Specification of terminal conditions is discussed below. Resolution of the model duration question is well beyond the scope of this effort. The questions of time preference and uncertainty are intimately related. Resolution of the question, however depends upon the problem at hand. Theoretical investigation may be found in Arrow and Kurtz; Graff; and Boussard. Also, a review of the literature may be found in Nuthall.

### **8.1.3 Time Interval Representation**

A discrete time representation is used in all models in this book. Within such a setting one must define time sub-intervals, hereafter called periods. One may specify periods of uniform or non-uniform duration. Basically, the specification depends upon the nature of the decisions to be made and the length of the production cycle.

Annual time disaggregation is most common. However, decades, half years, quarters and months have been used depending upon the problem structure. Model size usually increases rapidly as more periods are specified. Further, it is possible to have fine breakdowns in resource availability within time periods, for example, having weekly detail on some variables (e.g., product scheduling), but coarser detail on others (i.e., machinery investment). The breakdown used must be determined by the elasticity of substitution ( $\sigma$ ) between resources at different times; i.e., when labor in two weeks is perfectly substitutable ( $\sigma=1$ ) then include these weeks in one larger period. Otherwise, multiple periods are defined.

### **8.1.4 Initial and Terminal Conditions**

Often initial and terminal conditions are specified when formulating a multi-period dynamic model. In such a model it is important to adequately reflect the in-process inventory and previously made commitments (e.g., trees planted in the past, animals on hand, or partially depreciated investments) as initial conditions.

Most models do not contain either an infinite time horizon or conditions where all dynamic enterprises stop at the end of the horizon. Consequently, terminal conditions are important. Terminal conditions reflect the value of in-process inventory beyond the final period explicitly modeled and should either value or require a minimum level of inventory. When used, terminal values should reflect the net present value of the future income

stream earned by ending the time horizon with a unit of the in-process inventory. Such conditions can insure that model activity will be reasonable up until the final year. The error created by ignoring terminal conditions can be illustrated through example. Suppose a firm uses trucks which must be replaced after 6 years. Given a model with a 7 year horizon and no terminal conditions, there would be little incentive to buy new trucks toward the end of the model time horizon, say in the 6th year. The model would simply not reflect gains from the future availability after the 7 year horizon. Thus in a seven year model, a new truck acquired in the 6th year would need a terminal condition reflecting the returns to having the truck in its 5 years of existence after the explicit model time frame. The specification of terminal values can be difficult, particularly in the case of investments which enhance production and therefore alter the income stream.

Initial and terminal conditions may be specified in several ways, as always dependent upon the problem. The most common specification of initial conditions has the initial quantities specified as an exogenous limit (i.e. that there are 40 acres of existing trees that are 20 years old). Initial conditions may also allow inventory to be purchased at a price or according to a price quantity schedule. The same basic modeling alternatives are available for depicting terminal conditions. The terminal in-process inventory may be required to equal or exceed a fixed quantity, valued at a fixed price, or valued according to a price quantity demand schedule. It is usually improper to omit terminal conditions. Terminal values sometimes can be inferred by observing, for example, the model shadow prices on intermediate inventory in early time periods. Terminal values are much more difficult to specify than initial conditions (due to the element of forecasting involved). However, these parameters are key to the meaningful modeling of multi-period dynamic situations.

### **8.1.5 Enterprise Life**

Enterprise life refers to the number of years that an activity lasts and may be assumed to be known or unknown. If enterprise life is unknown and is to be determined in the model, then constraints and activities must be present in the model to keep track of age of the items on hand. If not, a single activity can be used. Models using alternative enterprise life assumptions are given below.

### **8.1.6 Time Preference**

Time preference is an important dynamic modeling concern. Given that returns and costs occur over time and alternative streams must be compared, these must be placed on common footing, usually using the concepts of discounting and present value. The present value of a dollar earned in year  $t$ , given a constant opportunity cost of funds of  $r$  is  $(1+r)^{-t}$ . Thus, the current value of a flow of funds which varies from year to year ( $c_t$ ) over  $T$  years is

$$PV = \sum_{t=0}^T \frac{C_t}{(1+r)^t}$$

Once time preference is considered, the question of specifying the appropriate cost of funds, (also called the discount rate) arises. Many considerations enter, such as whether the rate should be in real or nominal terms, how to determine the opportunity cost of capital, how to include the cost of borrowing funds, how to consider the value of alternative investments, and how to take risk into account. It is beyond the scope of this book to cover how to determine such a rate. Rather, the reader is referred to the discussion in Bussey relative to rates in private firms or Mishan for discussion of rates in social choice situations.

### 8.1.7 Risk

Risk is definitely a factor in dynamic situations. Risk is almost always present as situations evolve over time. This chapter only treats certainty models and unique aspects of risk in the context of dynamics; The majority of the risk techniques which have been used are discussed in the risk programming chapter below.

## 8.2 Dynamic Linear Programming

The most straight forward linear programming model of a dynamic situation contains multi-year enterprises of known life with multiple years modeled. This is where our discussion of dynamic models will begin. Subsequently, models with unknown life will be discussed, followed by sections on the equilibrium, typical single period model with known and unknown life. For economy of description, we will refer to multi-period models as disequilibrium models, and typical period representations as equilibrium models.

### 8.2.1 Disequilibrium - Known Life

Consider a problem involving decisions on how to commit resources over a number of time periods considering items which, once begun, will require resources for a known number of periods. A formulation of this problem must consider resource availability, choice of variables and continuing resource usage during the life of multi-period enterprises. Time preference of income and initial and terminal conditions must also be incorporated. Assume all enterprises have a K year life so that an enterprise undertaken in year t lasts until t+K. A model of this situation, assuming resource use, yields and returns are independent of the year in which the activity begins is

$$\begin{aligned}
\text{Max} \quad & \sum_{t,j} (1+r)^{-t} C_{jt} X_{j,t} - \sum_{e < K} (1+r)^{-T} F_{je} I_{je} \\
\text{s.t.} \quad & \sum_{e,j} A_{ije} X_{j,t+e} \leq b_{it} \quad \text{for all } i \text{ and } t \\
& X_{j,t+e} \geq X_{j,t}^* \quad \text{for all } e > 0 \text{ and all } j \\
& \sum_j X_{j,T+e} \leq I_{je} \quad \text{for all } e < K \text{ and all } j \\
& X_{j,t} \geq 0 \quad \text{for all } j, t \text{ and } e < K
\end{aligned}$$

Here  $t$  is the year identifier and begins at zero for the first year;  $T$  is the last year in the model;  $j$  identifies alternative variables;  $e$  indexes the elapsed age of an enterprise;  $K$  is the life of an enterprise (i.e., the number of periods after starting the enterprise that resources are used);  $i$  indexes resources;  $r$  is the discount rate;  $C_{jt}$  is the profit from initiating variable  $X_{jt}$  in year  $t$ ;  $X_{jt}$  is the number of units of alternative  $j$  initiated in year  $t$ ;  $F_{je}$  is the terminal value of  $X_{je}$  after the explicit time period in the model lapses;  $I_{je}$  is the number of units of  $X_j$  which are  $e$  years old at the model completion;  $A_{ije}$  is the usage of resource  $i$  by the  $j$ th alternative when it is  $e$  years old;  $b_{it}$  is the endowment of resource  $i$  in year  $t$ ; and  $X_{j,t-p}^*$  is the initial condition giving the amount of  $X_j$  done  $p$  years before the model begins.

The model maximizes net present value of activity from years 0 through  $T$  plus the terminal value of in-process inventory. Annual profits are converted to present value by multiplying through by the discount factor. The model assumes the profits ( $C_{jt}$ ) are collapsed into the period when the  $X_j$  variable is initiated. This is done by calculating the term  $C_{jt}$  as

$$C_{jt} = \sum_e (1+r)^{-(t+e)} h_{je}$$

where  $e \in \{K-t, T-t\}$

where  $h_{je}$  is the net profit from  $X_{jt}$ . When it is  $e$  years old. The inequality for  $e$  accounts for cases where the life of the enterprise extends beyond the time horizon.

The second term of the objective function gives the terminal value of enterprises not entirely completed during the years the model explicitly covers. For example, enterprises begun in the last year explicitly modeled are valued at  $F_{j1}$ , since they will be one year old in the period following the model. The  $F_{je}$  terms depict a future stream of profits to activities lasting beyond the model horizon discounted back to year  $T$ .

The first model constraint requires resource use to be less than or equal to resource availability in that year. Annual resource use depends upon enterprises begun in the current time period as well as items which began up to  $K$  years in the past. Thus, the summation subscripting adds all variables which are of age

0 to K years old in the current year (t). Resource use is assumed to depend only on the elapsed age (e) of the activity. Each alternative (j) once committed will always exist for years t+0 to t+K using an a priori specified amount of resources ( $A_{ije}$ ) each year. The next constraint set specifies the initial conditions: i.e., the amount of enterprises undertaken before the model was started which use resources during the time period covered by the model. The last constraint set adds up in process inventory at the end of the explicit time horizon setting the variable  $I_{je}$  equal to the amount of  $X_j$  which is e years old in the final model year and will extend into future years, thus the e=K case is not included.

The formulation assumes all enterprises have the same life. This clearly is not necessary and will be relaxed in the example. The assumption could be relaxed in the general model by adding a subscript j to the e parameter which would depict the life of the jth activity. The model also assumes equal length periods.

### 8.2.1.1 Example

Suppose a farmer wishes to establish a five-year crop plan. Two crops are under consideration: wheat and strawberries. Constraints restricting the choice of plans include quantities of land and water. Assume wheat has a one year life and requires one acre of land while an acre uses \$60 of variable costs per acre using 1 acre-ft/acre of water and yielding 100 bushels of wheat which sell for \$4.00 each. Assume strawberries have a 3 year life.

The Strawberry costs, resource usage and yields are

	<u>Year 0</u>	<u>Year 1</u>	<u>Year 2</u>
Cost/Acre	150	280	300
Yield/acre in tons	0	7	7
Water/acre in acre-feet	.8	4.5	4.5

The price per ton of strawberries is \$140. The farm has 700 acres, 50 planted in 0 year old strawberries and 10 in 1

year old strawberries. Water available consists of 1200 acre ft. per year.

We need terminal conditions to value those items which are carried into the fifth and beyond years.

Assume that the following values have been derived.

Product Terminal Value	
New strawberries (0 years old the year after the model)	\$160/acre
1 year old strawberries	\$110/acre

The initial tableau of this formulation appears in Table 8.1. The tableau reflects the known life assumption in that, under the strawberry activities, when the strawberries are begun, they commit resources both

in the year they are begun and in later years. Thus, while one acre of land in year one is used for year zero strawberries, it is also matched by an acre in years one and two. This reflects the assumption that strawberries are always kept three years, i.e. that they have a known life. The initial conditions on the problem are reflected by the right-hand sides. Notice the year one land constraint has a resource availability of 640, which is the 700 acres of available land less the sixty acres previously planted to strawberries. The water constraint is similarly adjusted. This is also reflected in the second year where the resource endowments reflect continuation of the ten acre patch previously existing. Terminal conditions in the model are reflected on the strawberries begun in years three and four, which are not fully completed during the model time frame.

The solution is shown in Table 8.2. The GAMS file DISEQK depicts this problem. The solution reflects the disequilibrium nature of the model. Wheat acreage varies from year to year as the acres of strawberries are established. Note that no strawberries are established in the last couple of years, and therefore there are no terminal quantities of strawberries. This is due to relatively low terminal conditions on strawberries. The solution indicates that it would cost as much as \$330 in year five to plant strawberries. This means that the terminal conditions would have to be at least \$490 before it would be optimal in the model to do that.

### 8.2.2 Disequilibrium - Unknown Life

A second possibility involves modeling of situations wherein the exact life of activities is to be endogenously determined (although a maximum life is known) in a multi-period context. A formulation of this problem must include the considerations in the known life model as well as variables depicting the decision on whether to continue or terminate an activity. Adopting the same assumptions as above, a general formulation of this model is

$$\begin{aligned}
 \text{Max} \quad & \sum_{t,j,e} (1+r)^{-t} C_{je} X_{j,t,e} - \sum_{e,j} (1+r)^{-T} F_{je} I_{je} \\
 \text{s.t.} \quad & \sum_e A_{ije} X_{j,t,e} \leq b_{it} \quad \text{for all } i \text{ and } t \\
 & X_{j,0,e} \leq X_{j,0,e} \quad \text{for all } j \text{ and } e < K \\
 & X_{j,T,e} \leq 0 \quad \text{for all } j \text{ and } e < K \\
 & X_{j,t+1,e+1} \leq X_{j,t,e} \quad \text{for all } t, j \text{ and } e > 0 \\
 & X_{j,t,e} \geq 0 \quad \text{for all } j, t, e
 \end{aligned}$$

Where  $t$  is the time index;  $T$  is the length of the total planning horizon;  $j$  is an index for identifying alternative



variables;  $e$  is an index identifying the elapsed age of a variable;  $K$  is the maximum age of a variable;  $i$  is an index which identifies resources;  $r$  is the discount rate;  $C_{je}$  is the per unit profit from variable  $j$  when it is of age  $e$ ;  $X_{jte}$  is the number of units of alternative  $j$  on hand in period  $t$  which are of elapsed age  $e$ ;  $F_{je}$  is the terminal value of incomplete units of  $X_j$  which are of elapsed age  $e$ ;  $A_{ije}$  is the resource  $i$  usage by one unit of the production represented by variable  $j$  when it is of elapsed age  $e$ ;  $b_{it}$  is the endowment of resource  $i$  in period  $t$ ; and  $X_{joe}$  is the initial amount of enterprise  $j$  of elapsed age  $e$  before the model begins (in time period 0).

This model has many common features with the known life model. In particular, the objective function interpretation is virtually identical. However, there are several distinctive features which should be brought out.

Since the life of a production variable may be terminated, the decision variable now is  $X_{jte}$  which allows for the possibility of employing variable  $j$  in year  $t$  when it is of elapsed age  $e$ . The constraint before the nonnegativity conditions relates the amount of  $e$  year old variable in year  $t$  to the amount of  $e-1$  year old variable in year  $t-1$ . The inequality on this constraint permits the predecessor to be nonzero and the successor to be zero (thus depicting selections less than maximum life for the variable).

This model is probably more realistic than the earlier one, but this has a price. The number of variables has been multiplied by the number of years a production activity could exist ( $K + 1$ ) and the number of constraints has expanded. This is potentially a much larger model.

#### 8.2.2.1 Example

For this example we use the data above plus longer retention of strawberries. Assume they may be kept up to 4 years and that in the fourth year the planting costs \$400 with the yield being 5, and water use being 5.7 acre feet. We will also assume that the terminal value of 3-year old strawberries is \$20/acre.

The resultant model is given in Table 8.3. The model shows several things. First the initial endowments of strawberries, are reflected on the right-hand side in the first year. Here there are 50 acres of zero year old strawberries, 10 of one year old strawberries, and zero of two year old strawberries. Second, the unknown life assumption is reflected in the linkage constraints that are labeled straw 0-1 year one etc. These constraints require that the strawberries in year one that are zero years old be a prerequisite for strawberries that are one year old in the second year, and similarly two year old strawberries in year two are a prerequisite for three year old strawberries in year three. Third, the model thus has its choice of whether to continue or not continue the strawberries throughout their life. This is reflected in the solution (Table 8.4), which shows that three year old strawberries are never used, indicating their life is always terminated, except that the two year old strawberries

are kept in the terminal condition in the final year. Fourth, disequilibrium behavior is reflected in that different amounts of strawberries are established in each year, as are different amounts of wheat.

### 8.2.2.2 Comments

An alternative form of the disequilibrium unknown life model merges the concepts of the known and unknown life models. In this model the linkage equations above are dropped and the X variables are assumed to have known life wherein the alternative variables are formulated for each possible activity age. Thus in our example, we would have first year variables for strawberries planted in the first year and kept with a known life of 1, 2, 3, and 4 years (4 variables). Such an example will be illustrated in the equilibrium unknown life section.

### 8.2.3 Equilibrium - Known Life

Disequilibrium models are usually relatively large. Further, the specification of the initial and terminal conditions is of key importance. However, such conditions can be difficult to specify. Equilibrium models provide an alternative approach. In an equilibrium model, the firm is assumed to operate in a steady state manner, repeatedly making identical decisions year after year. In such a case, initial and final inventories of in-process goods will be equal; thus, the initial and final conditions may simply be set equal and their optimal levels determined by the model. This leads to a smaller model depicting a representative equilibrium year. Equilibrium solutions, however, do not portray growth situations or time paths of adjustment; only final equilibriums are created.

Equilibrium models are developed as follows: assume we have a variable with life of 4 periods and resource use, yield, etc., equal to  $A_e$ , where  $e$  is the elapsed age of the activity (0-3). Let us (assuming we start with zero initial activity) portray the resource use over several periods.

		Begin Activity in Period					
		1	2	3	4	5	6
Period 1	resource use	$A_0$					
Period 2	resource use	$A_1$	$A_0$				
Period 3	resource use	$A_2$	$A_1$	$A_0$			
Period 4	resource use	$A_3$	$A_2$	$A_1$	$A_0$		
Period 5	resource use		$A_3$	$A_2$	$A_1$		
Period 6	resource use			$A_3$	$A_2$	$A_0$	
Period 7	resource use				$A_3$	$A_1$	$A_0$

Once the activity pattern enters period 4 we have resource use by each age of the activity. In this and subsequent periods, resource use in any period may be written as

$$R_t = A_3 X_{t-3} + A_2 X_{t-2} + A_1 X_{t-1} + A_0 X_t$$

where  $X_t$  is the quantity of the activity begun in period  $t$ .

Under an equilibrium assumption the same thing is done each year in a continually repeated sequence.

Thus,  $X_t = X_{t-1} = X_{t-2} = X_{t-3}$  and the resource use may be rewritten as

$$R_t = A_3 X_t + A_2 X_t + A_1 X_t + A_0 X_t = (A_0 + A_1 + A_2 + A_3) X_t = X_t E$$

In an equilibrium model, then, the resource use for a variable is the sum of its resource usages over its whole life.

(At this juncture we should observe that these assumptions imply that initial and final inventories are equal.)

Thus, a general formulation of the equilibrium model with known life is

$$\begin{aligned} \text{Max } & \sum_j C_j X_j \\ \text{s.t. } & \sum_j A_{ij} X_j \leq b_i \text{ for all } i \\ & X_j \geq 0 \text{ for all } j \end{aligned}$$

where:  $X_j$  is the quantity of the  $j$ th enterprise produced in equilibrium.

$C_j$  is the revenue per unit of  $X_j$  and equals the sum of the returns to the activity over the periods ( $e$ ) of its life, which is assumed to be known;

$A_{ij}$  is the use of resource  $i$  per unit of  $X_j$  and equals  $A_{ije}$  or the sum of the resource usages over the years of the enterprise's life; and

$b_i$  is the amount of resource  $i$  available.

### 8.2.3.1 Example 1

Two examples will be shown for this particular formulation to illustrate its features. The first continues the example used throughout the chapter; in the second, initial and final inventory are explicitly handled.

The equilibrium known life nature of this model is reflected in Table 8.5 and GAMS formulation EQUILK. The model reflects the known life assumption under the strawberry activity in that the objective function of the strawberry activity equals the sum of the objective function values over all three years of its life. Similarly, land use equals three, indicating that when one acre is started in the first year that implicitly an acre is committed a year from now and two years from now. Under the equilibrium assumption for each acre started now, there will be two additional acres; one that was started one year ago and one that started two years ago, for a total of three acres of land. Initial and terminal conditions are not reflected. The solution to this model is given in Table 8.6, which shows that the equilibrium solution is to plant 479 acres of wheat each and every year and 74

acres of strawberries. This model can also be re-expressed in terms of average resource use. This is done in Table 8.7, where an average of one acre of land is used every year generates an average of \$410 and the usage of 3.27 acre feet of water. The solution for this model essentially identical to the solution for the previous model, but the strawberry variable equals 221. This indicates that the equilibrium solution averages 221 acres of strawberries. Thus, in the strawberry rotation, one-third of the 221 acres (or 74 as in the earlier model) would be first year, one-third second year and one-third year.

### 8.2.3.2 Example 2

The above example does not explicitly show the initial and final inventory situations. Therefore, we wish to construct another example that demonstrates this point more explicitly.

Suppose a farmer wishes to establish a crop rotation. Two crops are planted: corn and soybeans. Assume crop yield varies with dates of planting and harvest, thus the yields are dependent upon time of planting. Time is disaggregated into five annual periods: one in the fall after harvest through spring before planting, two during planting, and two during harvest. Cultural operations of the crop are plowing, planting and harvesting. Plowing may be done any time after a crop is harvested (during or after the harvest period) and before planting (during or before the planting period). Also assume that corn yields depend upon whether corn follows corn or soybeans. Land and labor are the limiting resources.

Technical data pertinent to the model are:

	<u>Corn</u>	<u>Soybeans</u>
Prices/Unit	\$2.50	\$6.50
Production Cost/ Acre	100	50
Labor Use in hrs/acre		
Plowing	.4	.3
Planting	.15	.15
Harvesting	.35	.17

Yields:

		Corn after Corn Planting Period		Corn After Soybeans Planting Period		Soybeans (after Anything) Planting Period	
		Pd2	Pd3	Pd2	Pd3	Pd2	Pd3
Harvest	Pd4	130	120	145	133	35	45
Period	Pd5	125	110	137	129	33	42

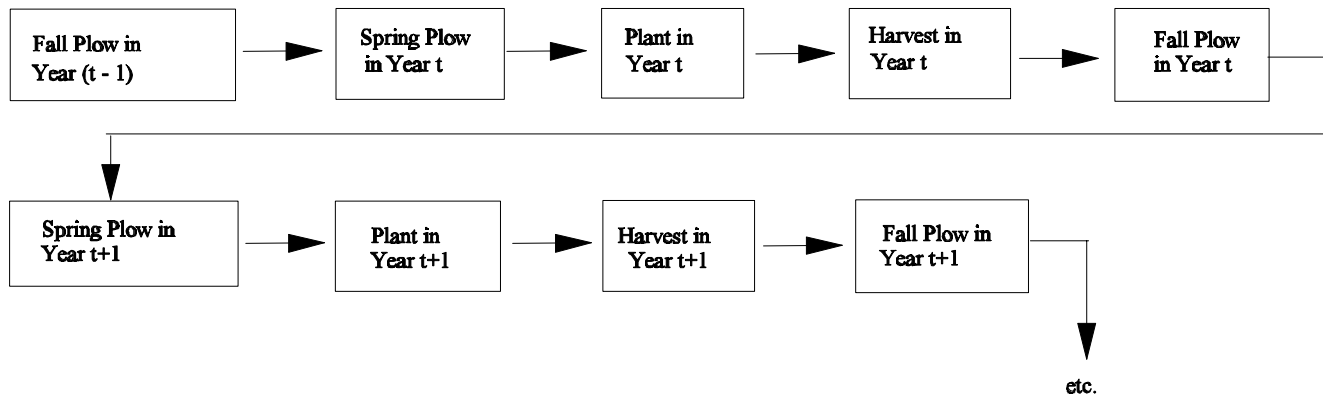
Labor Availability (hrs)

<u>Period</u>	<u>Available Labor</u>
Post harvest/Preplant (Pd1)	80
Plant (Pd2)	65
Plant (Pd3)	75
Harvest (Pd4)	100
Harvest (Pd5)	80

The farm has 400 acres.

Before formulating this model, its dynamic nature should be explored. Specifically, we must address this question: what are the items which constitute the initial and terminal conditions?

A diagram of the dynamic process appears below:



This dynamic process reflects linkages between last year's fall plowing with this years activities. The equilibrium assumption then is that this year's fall plowing and crop mix equals last year's. Therefore, we will require this to be true by using only one set of variables for fall plowing which by nature assumes equality. The

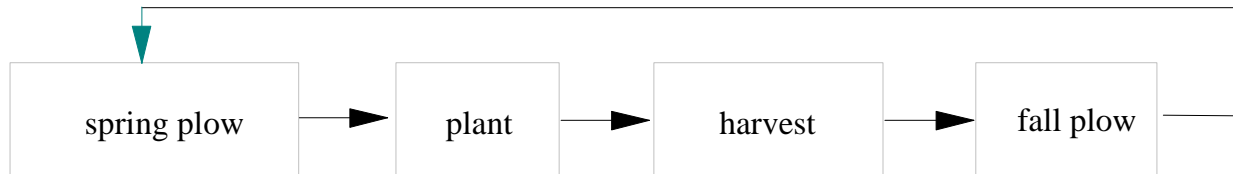


diagram of the resulting dynamic process is which implies that planting precedes harvest which precedes planting; thus, planting precedes itself in an assumed equilibrium situation.

The model formulation' reflecting this is shown in Table 8.8. Note that sequencing is modeled as illustrated earlier. This introduces the dynamic equilibrium assumption, in that plowing precedes planting, which precedes harvest, which precedes plowing, making the timing circular where the functions both precede themselves and follow themselves. The solution to this model is indicated in Table 8.9, and shows a continuous rotation with 200 acres of corn are planted after soybeans and 200 acres of soybeans are planted after corn.

### 8.2.3.3 Comments

Several aspects of the equilibrium model require discussion. First when a model reaches an equilibrium, the resources available must be in equilibrium. Thus, the same amount of resources must be available on a continuing basis. Similarly, resource use of a variable of a given age must be the same year after year. Such assumptions rule out the use of this model when examining cases with depletable resources.

A second assumption involves the word repeatability. Namely, the plan at any point in time must be repeatable. Given an alternative which lasts  $n$  periods the solution will have  $1/n$  units of the alternative in each stage of its life. In the wheat strawberry  $1/3$  of the acreage is in each of the three years of the strawberries life. Further, in the corn-soybean example, no more than  $1/2$  the acreage will be devoted to corn following soybeans and an equal acreage must be devoted to soybeans followed by corn.

A final line of comments relates to discounting. These authors have not examined discounting at great length, but feel that discounting should not be included in this model since the return in every year is identical.

### 8.2.4 Equilibrium - Unknown Life

A version of the equilibrium model may be formulated in which the multi-period activities are of unknown life. The assumption is retained that the same thing is done each period. Thus, resource use in a typical period is summed across the possible ages that an activity may be kept. A general formulation of the model is

$$\begin{array}{ll}
 \text{Max} & \sum_j \sum_e C_{je} X_{je} \\
 \text{s.t.} & \sum_j \sum_e A_{ije} X_{je} \leq b_i \quad \text{for all } i \\
 & X_{j,e} \leq X_{j,e-1} \quad \text{for all } j \text{ and } e > 0 \\
 & X_{je} \geq 0 \quad \text{for all } j \text{ and } e
 \end{array}$$

where:  $X_{je}$  is the quantity of the  $j$ th activity produced and kept until it is  $e$  periods old;

$C_{je}$  is the per unit returns to  $X_{je}$ ;

$A_{ije}$  is the per unit usage of resource  $i$  by  $X_{je}$ ;

$b_i$  is the amount of resource  $i$  available;

The model maximizes profits subject to constraints on resource use and age sequencing. The age sequencing constraints state that the amount of enterprise  $j$  of age  $e$  must be less than or equal to the amount of that enterprise that was kept until age  $e-1$ .

#### 8.2.4.1 Example

A tableau of the unknown life equilibrium model in the wheat, strawberry example context is given in Table 8.10. There we again have the structure where first year strawberries are required before second year strawberries can be grown. Similarly, two year old strawberries are required before three year old strawberries, etc. The solution, which is given in Table 8.11, indicates that the model chooses to terminate the life of strawberries at the end of three years and that it would cost \$88 to continue them into the fourth year. The solution also is indicative of what terminal conditions would be needed had we set them appropriately in the equilibrium model, where the terminal value of zero year old strawberries is around \$488 and the terminal value of one year old strawberries is \$164 and the terminal value of two year old strawberries is \$0. A slightly more compact formulation may be constructed in which the activities represent the activity kept for  $e$  years. This model is illustrated in Table 8.12.

### 8.2.5 Overall Comments on Dynamic Linear Programming

Dynamic LP may be used in many settings. The models, however, do involve numerous assumptions and require different computational and data resources. The disequilibrium models in particular become quite large when used and also require constructing "good" terminal conditions. Equilibrium models are more compact, yet use assumptions which may be unrealistic given a problem situation.

Further, both models suffer from a "curse of certainty"; clearly the optimal solution depends upon the assumed current and future conditions. However, the model can include uncertainty. Kaiser and Boeljhe (1980) included uncertainty in the disequilibrium model using the expected value variance and MOTAD formulations (although their MOTAD approach is wrong). Yaron and Horowitz (1972B) incorporated risk via a multi-stage discrete stochastic method.

Three types of models have been used in many research studies. In terms of equilibrium - known life models, Swanson and Fox (1954); and Peterson (1955) present early models. Throsby (1967) formalized the approach. In fact, the use of average statistical budgets in a single year model assumes equilibrium (e.g., the models in Heady and Srivistava [1975]). Brandao, McCarl and Schuh (1984); and El Nazer and McCarl (1986) present models which apply the equilibrium model. Equilibrium - unknown life models are not as easily found, although El Nazer and McCarl (1986) and McCarl et al. (1977) provide examples.

In terms of disequilibrium models, early models with known life come from Swanson and Fox (1954) and Dean and deBenedictus (1964) while White et al. (1978); and Spreen et al. (1980) provide later examples. Unknown life type models are given in Loftsguard and Heady (1959); and Candler (1960) while Irwin (1968) provides a review article and Boussard (1971); Norton, Easter and Roe (1980); and Reid, Musser and Martin (1980) provide other examples and references.

### **8.3 Recursive Programming**

Another LP formulation dealing with dynamic situations is known as recursive programming (Day, 1963). Basically, recursive programming models involve problems in which model coefficients are functionally dependent upon earlier model solutions and an exogenously specified time path. Following Day and Cigno, a recursive programming model consists of a constrained optimization model; and a data generator. The data generator given the solution in period  $t$ , prepares the input for period  $t+1$  including defining a set of constraints which relates the feasible values of current variables to past values of variables and exogenous events. The model then is optimized for each time period with the data generator updating the model to the next time period.

While recursive programming does not have to be cast as a LP, the LP version will be presented. The general recursive LP sequentially maximizes an LP for each of a number of periods.



$$\begin{aligned}
\text{Max for period } t \quad Z_t &= \sum_j C_{jt} X_{jt} - \sum_k d_k Y_{kt} \\
\text{s.t.} \quad & \sum_j A_{ijt} X_{jt} - \sum_k E_{ik} Y_{kt} \leq b_{it} \quad \text{for all } i \\
& X_{jt} \geq 0, \quad Y_{kt} \geq 0 \quad \text{for all } j \text{ and } k
\end{aligned}$$

Where:  $C_{jt}$  are objective function parameters functionally dependent upon the previous objective function parameters ( $C_{j,t-1}$ ), lagged optimal decision variables ( $X_{j,t-1}$ ,  $Y_{k,t-1}$ ), and exogenous events;  $X_{jt}$  are the values of the decision variables at time  $t$ ;  $A_{ijt}$  are the resource  $i$  usages by  $X_{jt}$  functionally dependent upon lagged values,  $d_k$  are objective function coefficients which are stable over time;  $Y_{kt}$  are the optimal values of the  $k$ th  $Y$  variable in time period  $t$ ;  $E_{ik}$  are the usages of resource  $i$  which do not change over time;  $b_{it}$  are the resource  $i$  limits, functionally dependent upon lagged phenomena.

The recursive model is not solved over all time periods simultaneously, rather an optimal solution is constructed for each time period. There is no guarantee (or necessarily a desire) that these solutions are optimal over all periods. Rather, the solution represents an adaptive stream of decisions.

We have not specified the functional form of the lagged functions as they most often depend upon the problem at hand.

### 8.3.1 Example

A simple recursive programming example arises from a "cobweb" type model. Assume we have a group of producers who choose between two crops based on last year's price. Further assume that producers dampen their adjustment so that production varies only as much as 2 percent from last year's acreage. Production costs are \$135 per acre for Crop 1 and \$85 per acre for Crop 2. The yields are Crop 1 - 130 bu/acre; Crop 2 - 45 bu/acre. Assume the typical producers each have 600 acres. Suppose, last year's acreage mix was 50 percent Crop 1 and 50 percent Crop 2 (300 acres of each for the average producer). Lastly, assume the demand curve for Crop 1 in terms of this producers production price dependent form is  $P_1 = 20 - .00045 Q_1$  and the demand curve for Crop 2 is  $P_2 = 10.9 - .00045 Q_2$  where  $Q$  is the quantity produced. Such a situation leads to the following programming problem.

$$\begin{array}{rcl}
\text{Max } Z_t & = & (130 P_{1t} \& 135) X_{1t} \quad \% \quad (45 P_{2t} \& 85) X_{2t} \\
& & X_{1t} \quad \% & X_{2t} \quad \# & 600 \\
& & X_{1t} & \# & 1.02 X_{1t\& 1} \\
& & X_{1t} & \$ & .98 X_{1t\& 1} \\
& & & X_{2t} \quad \# & 1.02 X_{2t\& 1} \\
& & & X_{2t} \quad \# & .98 X_{2t\& 1} \\
& & X_{1t}, & X_{2t} \quad \$ & 0
\end{array}$$

where:  $P_{1t+1} = 20 - 0.00045 Q_{1t}$

$$Q_{1t} = 130 X_{1t}$$

$$P_{2t+1} = 10.9 - 0.00045 Q_{2t}$$

$$Q_{2t} = 45 X_{2t}$$

The recursive solution to this model over 6 periods is

Time Period	$X_{1t}$	$X_{2t}$	$P_{1t+1}$	$P_{2t+1}$	$Z_t$
0	300.000	300.000	2.450	4.825	----
1	306.000	294.000	2.099	4.703	94995.750
2	311.880	288.120	1.755	4.584	79491.454
3	306.118	293.882	2.092	4.701	64163.394
4	311.995	288.005	1.748	4.582	79182.851
5	306.235	293.765	2.085	4.699	63860.831
6	312.110	287.890	1.742	4.580	78874.189

### 8.3.2 Comments on Recursive Programming

Recursive programming models explicitly represent lagged adjustment. Several forms of lagged adjustment have been commonly used. A frequently used form involves "flexibility constraints" in which a variable is allowed to vary at most from its predecessor by a percentage (as done in the example). Day (1963) first used this form of constraint; Schaller and Dean did so later. Sahi and Craddock (1974, 1975) present information on estimation of these parameters.

Models involving lagged adjustment to price are also common. Day (1978) reviews the early literature involving the cobweb model and goes over a number of considerations involved in its use. The approach involved is similar to the example above where current price is a function of lagged production. Such an approach is common in recursive programming, and in fact has been used in other contexts (e.g., see the discussion of

iterative methods in Judge and Wallace (1958) and Tramel and Seale (1959,1963)). The recursive programming model generally adjusts in one-year sequences. Multi-year activities must then be converted to single-year activities by annualizing costs and putting in expected future returns. The literature does not cover considerations where resource use of an activity is expected to change over time, but they are possible.

Finally, we come to usage of recursive programming. Actually, the vast majority of the usages are centered in and around the work of Day. A review of applications is given in Day and Cigno and several other references appear in Judge and Takayama (1973).

## References

- Boussard, J. "Time Horizon, Objective Function and Uncertainty in a Multiperiod Model of Firm Growth." American Journal of Agricultural Economics. 53(1971):467-77.
- Brandao, E.S., B.A. McCarl and G.E. Schuh. "Predicting the Impact of New Cropping Practices upon Subsistence Farming; A Farm-Level Analysis in Brazil." Western Journal of Agricultural Economics. 9(1984):329-41.
- Bussey, L. The Economic Analysis of Industrial Projects. Prentice Hall, New Jersey. 1978.
- Day, R. "Cobweb Models with Explicit Suboptimization." In Modeling Economic Change: The Recursive Programming Approach. R. Day and A. Cigno (eds.) North Holland Publishing Company, Amsterdam. 1978.
- \_\_\_\_\_. Recursive Programming and Production. North Holland Publishing Company, Amsterdam. 1963.
- Day, R. and A. Cigno (eds.). Modeling Economic Change: The Recursive Programming Approach. North Holland Publishing Company, Amsterdam. 1978.
- Dean, G. and M. deBenedictus. "A Model of Economic Development for Peasant Farms in Southern Italy." Journal of Farm Economics. 46(1964):295-312.
- El-Nazar, T. and B.A. McCarl. "The Choice of Crop Rotation: A Modeling Approach and Case Study." American Journal of Agricultural Economics. 68(1986):127-136.
- Heady, E. and V. Srivistava. Spatial Sector Programming Models In Agriculture. Iowa State University Press, Ames. 1975.
- Hildreth, C. and S. Reiter. "On the Choice of a Crop Rotation." In Activity Analysis of Production and Allocation, T. Keepmans (ed.), John Wiley & Sons, 1951, pp. 177-88.
- Irwin, G. "A Comparative Review of Some Firm Growth Models." Agricultural Economics Research. 20(1968):82-100.
- Judge, G. and T. Takayama. Studies in Economic Planning Over Space and Time. North Holland Publishing, Amsterdam, 1973.
- Kaiser, E. and M. Boeljhe. "A Multiperiod Risk Programming Model for Farm Planning." North Central Journal of Agricultural Economics. 2(1980):47-54.
- Madigliani, F. and Cohen, K. The Role of Anticipations and Plans in Economic Behavior and Their Use in Economic Analysis and Forecasting. Studies in Business Expectations and Planing, No. 4, University of Illinois, 1961.
- McCarl, B.A., W.V. Candler, D.H. Doster, and P.R. Robbins. "Experiences with Farmer Oriented Linear Programming for Crop Farming." Canadian Journal of Agricultural Economics. 25(1977):17-30.
- Mishan, E. Cost Benefit Analysis. Praeger Press, New York, 1976.
- Norton, G., K. Easter, and T. Roe. "American Indian Farm Planning: An Analytical Approach to Tribal Decision Making." American Journal of Agricultural Economics. 62(1980):689-99.
- Nuthall, P.L. "A Survey of Methods for Determining a Planning Horizon." Review of Marketing and Agricultural Economics. 48(1980):21-36.
- Peterson, G. "Selection of Maximum Profit Combinations of Livestock Enterprises and Crop Rotation." Journal of Farm Economics. 37(1955):546-54.

- Reid, D., W. Musser, and N. Martin. "Consideration of Investment Tax Credit in a Multiperiod Mathematical Programming Model of Farm Growth." American Journal of Agricultural Economics. 62(1980):152-57.
- Sahi, R. and W. Craddock. "Estimating Crop Acreages in the Prairie Provinces: Application of Recursive Programming." Canadian Journal of Agricultural Economics. 23(1975):1-16.
- \_\_\_\_\_. "Estimation of Flexibility Coefficients for Recursive Programming Models - Alternative Approaches." American Journal of Agricultural Economics. 56(1974):344-50.
- Schaller, W. and G. Dean. "Predicting Regional Crop Production: Applications on Recursive Programming." USDA Technical Bulletin No. 1329, Washington, D.C. 1965.
- Spreen, T.H., B.A. McCarl, and T.K. White. "Modeling the Cattle Subsector: A Demonstration in Guyana." Canadian Journal of Agricultural Economics. 28(1980):17-32.
- Swanson, E. and K. Fox. "The Selection of Livestock Enterprises by Activity Analysis." Journal of Farm Economics. 36(1954):78-86.
- Throsby, C. "Dynamic Programming Activity Analysis and the Theory of the Firm." Review of Marketing and Agricultural Economics. 36(1968):20-27.
- Tramel, T. and A. Seal. "Reactive Programming of Supply and Demand Relations: Applications to Fresh Vegetables." Journal of Farm Economics. 41(1959):1012-22.
- \_\_\_\_\_. "Reactive Programming - Recent Development." In Interregional Competition Research Methods. R.A. King (ed.), University of North Carolina, Raleigh, 1963 pp. 59-68.
- White, T.K., B.A. McCarl, R.D. May and T.H. Spreen. "A Systems Analysis of the Guyanese Livestock Industry." Agricultural Systems. 3(1978):47-66.
- Yaron, D., and V. Horowitz. "Short and Long Run Farm Planning Under Uncertainty: Integration of Linear Programming and Decision Tree Analysis." Canadian Journal of Agricultural Economics. 20(1972):17-30.

Table 8.1. Disequilibrium Known Life Example													
Rows	Year 1		Year 2		Year 3		Year 4		Year 5		Terminal Conditions		RHS
	Wheat	Straw-berries	Wheat	Straw-berries	Wheat	Straw-berries	Wheat	Straw-berries	Wheat	Straw-berries	Aje 0	Aje 1	
Objective	340	1230	340	1230	340	1230	340	550	340	-150	160	110	Max
Land Year 1	1	1											#
Water Year 1	1	0.8											#
Land Year 2		1	1	1									#
Water Year 2		4.5	1	0.8									#
Land Year 3		1		1	1	1							#
Water Year 3		4.5		4.5	1	0.8							#
Land Year 4				1		1	1	1	1	1			#
Water Year 4				4.5		4.5	1	0.8					#
Land Year 5						1	1	1	1	1			#
Water Year 5						4.5	4.5	4.5	1	0.8			#
Strawberry 0											1		#
Strawberry 1												1	#

**Table 8.2. Disequilibrium Known Life Example Model Solution**

Objective = 1224296

Variables	Value	Reduced Cost	Equation	Slack	Shadow Price
Wheat Year 1	506.2	0	Land Year 1	0	340.0
Strawberries Year 1	133.8	0	Water Year 1	316.76	0
Wheat Year 2	539.9	0	Land Year 2	0	283.1
Strawberries Year 2	16.3	0	Water Year 2	0	56.9
Wheat Year 3	423.4	0	Land Year 3	0	336.9
Strawberries Year 3	126.6	0	Water Year 3	0	3.1
Wheat Year 4	557.1	0	Land Year 4	0	279.8
Strawberries Year 4	0	0	Water Year 4	0	60.2
Wheat Year 5	573.4	0	Land Year 5	0	340.0
Strawberries Year5	0	0	Water Year 5	57.05	0
Term Straw -0	0	0	Strawberry 0	0	-160
Term Straw -1	0	-8	Strawberry 1	0	-118

**Table 8.3 Disequilibrium Unknown Life Sample Problem**

Rows	Year 1			Year 2			Year 3			Year 4			Year 5			Terminal Conditions		RHS							
	Strawberries			Strawberries			Strawberries			Strawberries			Strawberries			Strawberries									
	Wheat	0	1	2	3	Wheat	0	1	2	3	Wheat	0	1	2	3	Wheat	0		1	2					
Objective	340	-150	700	680	300	340	-150	700	680	300	340	-150	700	680	300	340	-150	700	680	300	160	110	20	Max	
Land Year 1	1	1	1	1	1																			# 700	
Water Year 1	1	0.8	4.5	4.5	5.7																			# 1200	
Init Straw 0			1																					# 50	
Init Straw 1			1																					# 10	
Init Straw 2				1																				# 0	
Straw 0-1 Year 1		-1					1																	# 0	
Straw 1-2 Year 1			-1		1																			# 0	
Straw 2-3 Year 1				-1			1																	# 0	
Land Year 2						1	1	1	1	1														# 700	
Water Year 2						1	0.8	4.5	4.5	5.7														# 1200	
Straw 0-1 Year 2							-1					1												# 0	
Straw 1-2 Year 2								-1					1											# 0	
Straw 2-3 Year 2									-1					1										# 0	
Land Year 3											1	1	1	1	1									# 700	
Water Year 3											1	0.8	4.5	4.5	5.7									# 1200	
Straw 0-1 Year 3												-1												# 0	
Straw 1-2 Year 3														1										# 0	
Straw 2-3 Year 3															1									# 0	
Land Year 4																1	1	1	1	1				# 700	
Water Year 4											1	0.8	4.5	4.5	5.7									# 1200	
Straw 0-1 Year 4																	-1							# 0	
Straw 1-2 Year 4																		-1						# 0	
Straw 2-3 Year 4																			-1					# 0	
Land Year 5																				1	1	1	1	1	# 700
Water Year 5																				1	0.8	4.5	4.5	5.7	# 1200
Term Straw 0																									# 0
Term Straw 1																							1		# 0
Term Straw 2																									# 0



**Table 8.4. Disequilibrium Unknown Life Example Model Solution**

Objective = 1280757.0

Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
Wheat year 1	547.1	0	Land Year 1	0	340
Straw 0 year old year 1	92.9	0	Water Year 1	308.57	0
Straw 1 year old year 1	50.0	0	Init Straw 0	0	490
Straw 2 year old year 1	10.0	0	Init Straw 1	0	340
Straw 3 year old year 1	0	-40	Init Straw 2	0	0
Wheat year 2	557.1	0	Straw 0-1 Year 1	0	-490
Straw 0 year old year 2	0	-8	Straw 1-2 Year 1	0	-130
Straw 1 year old year 2	92.9	0	Straw 2-3 Year 1	10	0
Straw 2 year old year 2	50.0	0	Land Year 2	0	280
Straw 3 year old year 2	0	-322	Water Year 2	0	60
Wheat year 3	464.3	0	Straw 0-1 Year 2	0	-470
Straw 0 year old year 3	142.9	0	Straw 1-2 Year 2	0	-340
Straw 1 year old year 3	0	0	Straw 2-3 Year 2	50	0
Straw 2 year old year 3	92.9	0	Land Year 3	0	340
Straw 3 year old year 3	0	-40	Water Year 3	203.57	0
Wheat year 4	557.1	0	Straw 0-1 Year 3	0	-490
Straw 0 year old year 4	0	0	Straw 1-2 Year 3	0	-110
Straw 1 year old year 4	142.9	0	Straw 2-3 Year 3	92.85	0
				7	
Straw 2 year old year 4	0	0	Land Year 4	0	274
Straw 3 year old year 4	0	-349	Water Year 4	0	65.7
Wheat year 5	557.1	0	Straw 0-1 Year 4	0	-470
Straw 0 year old year 5	0	-330	Straw 1-2 Year 4	0	-360
Straw 1 year old year 5	0	0	Straw 2-3 Year 4	0	0
Straw 2 year old year 5	142.9	0	Land Year 5	0	340
Straw 3 year old year 5	0	-40	Water Year 5	0	0
Term Straw 0		0	Term Straw 0	0	-160
Term Straw 1		0	Term Straw 1	0	-110
Term Straw 2		0	Term Straw 2	142.86	-20

**Table 8.5. Equilibrium Known Life Example Formulation**

	Wheat	Strawberries		
Objective	340	1230		
Land	1	3	#	700
Water	1	9.8	#	1200

**Table 8.6. Equilibrium Known Life Example Solution**

Objective = 253441					
Variables	Value	Reduced Cost	Equation	Slack	Shadow Price
Wheat	479	0	Land	0	309
Strawberries	74	0	Water	0	31

**Table 8.7. Equilibrium Known Life Example Formulation with Average Activities**

	Wheat	Strawberries		
Objective	340	410		
Land	1	1	#	700
Water	1	3.27	#	1200

**Table 8.8 Rotation Example in Equilibrium Known Life Context**

Rows	Plant Corn										Plant Soybeans					RHS										
	Plow after corn					Plow after soybeans					after corn						after soybeans									
	pd1	pd2	pd3	pd4	pd5	pd1	pd2	pd3	pd4	pd5	after corn	p12	p13	hr4	hr5		after soybeans	p12	p13	hr4	hr5	after corn	p12	p13	hr4	hr5
Objective											225	200	263	233	223	178	243	178	243	223	178	243	165	243	223	Max
Land	1	1	1	1	1	1	1	1	1	1	213	175														# 400
Labor pd1	.4					.3					.2	.2	.2	.2	.2	.2	.2	.2	.2	.2	.2	.2	.2	.2	.2	# 80
Labor pd2		.4					.3				.2	.2	.2	.2	.2	.2	.2	.2	.2	.2	.2	.2	.2	.2	.2	# 65
Labor pd3			.4					.3			.4	.4	.4	.4	.4	.3	.3	.3	.3	.3	.3	.3	.3	.3	.3	# 75
Labor pd4				.4					.3		.4	.4	.4	.4	.4	.3	.3	.3	.3	.3	.3	.3	.3	.3	.3	# 100
Labor pd5					.4					.3	.4	.4	.4	.4	.4	.3	.3	.3	.3	.3	.3	.3	.3	.3	.3	# 80
plant after plow																										
corn pd2	-1	-1	-1	-1	-1						1	1				1	1				1	1				# 0
corn pd3	-1	-1	-1	-1	-1						1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	# 0
soybeans pd2						-1	-1	-1	-1	-1			1	1	1						1	1				# 0
soybeans pd3						-1	-1	-1	-1	-1			1	1	1						1	1				# 0
plow after harvest																										
corn pd4				1							-1	-1														# 0
corn pd5	1	1	1	1	1						-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	# 0
soyb pd4						1	1	1	1	1											-1	-1				# 0
soyb pd5						1	1	1	1	1											-1	-1	-1	-1	-1	# 0

**Table 8.9. Solution to Rotation Model**

Obj = 100777					
Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
Plow After Corn			Land	0	238
pd1	0	0	Labor pd1	80	0
pd2	87.5	0	Labor pd2	0	0
pd3	72.5	0	Labor pd3	16	0
pd4	0	-22	Labor pd4	0	56
pd5	0	0	Labor pd5	0	0
Plow After Soybeans			Plant After Plow		
pd1	0	0	corn pd2	128	0
pd2	0	0	corn pd3	0	238
pd3	0	0	soybeans pd2	0	0
pd4	0	-17	soybeans pd3	0	253
pd5	200	0	Plow After		
Plant corn after corn			corn pd4	200	0
pl2 hr4	0	-33	corn pd5	0	0
pl2 hr5	0	-26	soyb pd4	189	0
pl3 hr4	0	-58	soyb pd5	0	15
pl3 hr5	0	-63			
Plant Corn After Soybeans					
pl2 hr4	200	0			
pl2 hr5	0	-11			
pl3 hr4	0	-30			
pl3 hr5	0	-31			
Plant Soybeans After Corn					
pl2 hr4	0	-70			
pl2 hr5	0	-59			
pl3 hr4	189	0			
pl3 hr5	11	0			
Plant Soybeans After Soybeans					
pl2 hr4	0	-70			
pl2 hr5	0	-74			
pl3 hr4	0	-5			
pl3 hr5	0	-15			

Table 8.10. Equilibrium Unknown Life Example Formulation						
		Strawberries				RHS
Rows	Wheat	1	2	3	4	MAX
Objective	340	-150	700	680	300	
Land	1	1	1	1	1	# 700
Water	1	0.8	4.5	4.5	5.7	# 1200
Straw 1-2		-1	1			# 0
Straw 2-3			-1	1		# 0
Straw 3-4				-1	1	# 0

**Table 8.11. Equilibrium Unknown Life Example Solution**

Objective 253441					
Variables	Value	Reduced Cost	Equation	Slack	Shadow Price
Wheat	479	0	Land	0	309
Strawberries 1 year old	74	0	Water	0	31
Strawberries 2 year old	74	0	Straw 1-2	0	483
Strawberries 3 year old	74	0	Straw 2-3	0	232
Strawberries 4 year old	0	185	Straw 3-4	74	0

Table 8.12. Alternative Formulation of Equilibrium Unknown Life						
		Keep Strawberries				RHS
Rows	Wheat	1	2	3	4	MAX
Objective	340	-150	550	1230	1530	
Land	1	1	2	3	4	# 700
Water	1	0.8	5.3	9.8	15.5	# 1200

## CHAPTER IX: LINEAR PROGRAMMING MODELING: NONLINEARITIES AND APPROXIMATION

This chapter presents LP formulation techniques for representing nonlinear phenomena. The formulations fall into transformations and approximation classes. Transformations deal with minimization of the sum of absolute values; minimization of the largest absolute value; and maximization of a fraction. Approximations include grid point based formulations of problems and multi-variable functions.

### 9.1.1 Minimization of the Sum of Absolute Deviations

Suppose one wishes to minimize the sum of absolute deviations between a set of predictions and observations, where the predictions involve endogenously determined variables. Let the deviations be represented by:

where  $i$  identifies the  $i^{\text{th}}$  observation,  $d_i$  gives the deviation,  $Y_i$  an exogenously observed value,  $X_{ji}$  the exogenous data which go into forming the prediction of  $Y_i$ , and  $b_j$  the endogenous variable levels. The term  $Y_i$  minus the sum of  $X_{ji} b_j$  gives the difference between the observed level  $Y_i$  and its prediction given by  $(Y_i - \sum_j X_{ji} b_j)$ .

A LP constraint set is formed by moving the  $\sum_j X_{ji} b_j$  term to the left side of the equation.

$$d_i - \sum_j X_{ji} b_j = Y_i$$

The basic problem of minimizing the summed absolute values of all  $d_i$  is:

$$\begin{aligned} \text{Min } & \sum_i |d_i| \\ \text{s.t. } & d_i - \sum_j X_{ji} b_j = Y_i \quad \text{for all } i \\ & d_i \leq 0 \quad b_j \leq 0 \quad \text{for all } i \text{ and } j \end{aligned}$$

The variables in this formulation are  $d_i$  and  $b_j$ . The  $d_i$  are unrestricted in sign as are the  $b_j$ 's.

This problem is not a LP problem because of the nonlinear absolute value function. However, it can be transformed into a LP problem. First, we substitute for  $d_i$ , writing it as the difference of two non-negative variables:

$$d_i = d_i^+ - d_i^-$$

$d_i$  can take on negative values if  $d_i^+ > d_i^-$ ; conversely, if  $d_i^- > d_i^+$ , positive values result. The resultant problem

is

$$\begin{aligned} \text{Min } & \sum_i (s_i^+ + s_i^-) \\ \text{s.t. } & \sum_j X_{ji} b_j - Y_i = s_i^+ - s_i^- \quad \text{for all } i \\ & s_i^+, s_i^- \geq 0 \quad b_j \leq 0 \quad \text{for all } i \text{ and } j \end{aligned}$$

This problem is still nonlinear because of the absolute value term. However the absolute value terms can be simplified whenever either  $s_i^+$  or  $s_i^-$  equals zero as the consequent absolute value reduces to zero plus the other term. Algebraically, if the product of the deviation variables is zero, i.e.,

$$s_i^+ (s_i^-) = 0,$$

then the absolute value term can be written as the sum of the two variables

$$\begin{aligned} s_i^+ - s_i^- &= s_i^+ + s_i^- = s_i^+ + s_i^- \\ \text{whenever } & s_i^+ s_i^- = 0 \end{aligned}$$

Imposing the restriction that one or the other variable is zero, the formulation becomes

$$\begin{aligned} \text{Min } & \sum_i (s_i^+ + s_i^-) \\ \text{s.t. } & \sum_j X_{ji} b_j - Y_i = s_i^+ - s_i^- \quad \text{for all } i \\ & s_i^+ (s_i^-) = 0 \quad \text{for all } i \\ & s_i^+, s_i^- \geq 0 \quad b_j \leq 0 \quad \text{for all } i \text{ and } j. \end{aligned}$$

This is an LP formulation except for the constraint on the product of  $s_i^+$  and  $s_i^-$ . However, this constraint can be dropped. Consider a problem with only one observation  $Y$  without  $X$  and  $b$ . Under this case the formulation reduces to

$$\begin{aligned} \text{Min } & s^+ + s^- \\ \text{s.t. } & Y = s^+ - s^- \\ & s^+, s^- \geq 0 \end{aligned}$$

Rearranging the first constraint we obtain

$$s^+ = Y + s^-$$

In turn, tabling alternative values for Y (i.e., consider Y=4, Y=-6), including possible values of  $x_1, x_2$  and the resultant objective function sum yields

Y = 4			Y = -6		
$x_1$	$x_2$	$4x_1 + 6x_2$	$x_1$	$x_2$	$4x_1 + 6x_2$
4	0	4*	0	6	6*
16	12	28	14	20	34
Z + 4	Z	2Z + 4	Z	Z + 6	2Z + 6

\* These cases are the only ones in which  $4x_1 + 6x_2$  equals zero.

In the Y=4 case,  $x_1$  has to equal  $x_2 + 4$ . The left most part of the table gives several alternatives for this. The first is  $x_1=4$  and  $x_2=0$ , leading to a sum ( $4x_1 + 6x_2$ ) of 4. The second alternative (16 and 12) gives an objective function sum of 28. In general, for any choice for  $x_2 = Z$ , the  $x_1$  value must equal Z+4, and the objective function value becomes 2Z+4. Clearly, when 2Z+4 is minimized and Z is non-negative, the minimum occurs at Z=0, implying  $x_2=0$ . A similar conclusion can be reached for the negative Y case. Thus, minimization will automatically cause  $x_1, x_2$  to equal zero, and the nonlinear constraint is not necessary. Consequently the final formulation becomes

$$\begin{aligned} \text{Min } & \sum_i (c_i x_i) \\ \text{s.t. } & \sum_i a_{ij} x_i = \sum_j b_j Y_j \quad \text{for all } i \\ & x_i, x_j \geq 0 \quad b_j \leq 0 \quad \text{for all } i \text{ and } j, \end{aligned}$$

which is a linear program. This problem solves the original problem. The nonlinear problem has been transformed into an equivalent LP.

### 9.1.1.1 Example

Suppose a linear equation is to be fit predicting raw orange price as a linear function of the quantity of juice and fresh oranges sold given the following data:

Price of Raw Oranges	Quantity of Oranges Sold	Quantity of Juice Sold
10	8	5
5	9	1
4	10	9
2	13	8
6	15	2
9	17	3



Assume the prediction equation is  $Y_i = b_0 + b_1 X_{i1} + b_2 X_{i2}$ , where  $b_0$  is the intercept,  $b_1$  and  $b_2$  are the prediction parameters on the quantity of oranges and juice sold, respectively. Define  $X_{i1}$  and  $X_{i2}$  as the observations on the quantity of oranges and juice sold, respectively; and  $Y_i$  as the observed price. Suppose the desired criteria for equation fit is that the fitted data exhibit minimum sum of the absolute deviations between the raw orange price and its prediction. The formulation would be

$$\begin{aligned} \text{Min } & \sum_i d_i^+ + d_i^- \\ \text{s.t. } & d_i^+ - d_i^- = Y_i - b_0 - X_{i1} b_1 - X_{i2} b_2 \quad \text{for all } i \\ & d_i^+ \geq 0 \quad b_0, b_1, b_2 \leq 0 \quad \text{for all } i \end{aligned}$$

The equivalent LP formulation is

$$\begin{aligned} \text{Min } & \sum_i (d_i^+ + d_i^-) \\ \text{s.t. } & d_{i1}^+ - d_{i1}^- = 10 - b_0 - 8b_1 - 5b_2 \\ & d_{i2}^+ - d_{i2}^- = 5 - b_0 - 9b_1 - 1b_2 \\ & d_{i3}^+ - d_{i3}^- = 4 - b_0 - 10b_1 - 9b_2 \\ & d_{i4}^+ - d_{i4}^- = 2 - b_0 - 13b_1 - 8b_2 \\ & d_{i5}^+ - d_{i5}^- = 6 - b_0 - 15b_1 - 2b_2 \\ & d_{i6}^+ - d_{i6}^- = 9 - b_0 - 17b_1 - 3b_2 \\ & d_i^+, d_i^- \geq 0 \quad b_0, b_1, b_2 \leq 0 \quad \text{for all } i \end{aligned}$$

Moving the endogenous variables (i.e., the  $d_i^+$ 's and  $d_i^-$ 's) onto the left-hand side and substituting for the variables which are unrestricted in sign ( $b_0, b_1, b_2$ ) yields the final formulation given in Table 9.1. The GAMS formulation for this problem is called ABSOLUTE. The objective function minimizes the sum of the deviation variables subject to constraints relating the deviation variables to the difference between the observed and forecast levels for each observation. The coefficients on the intercept are plus ones; the coefficients on the other parameters ( $b_1, b_2$ ) are the observed levels. The right hand sides are the observed prices to be forecast.

The resultant solution yields an objective function value of 11.277, and the solution is shown in Table 9.2. The predictive equation yielded by this problem reveals that the price of oranges is predicted by the equation  $3.426 + (0.191 * \text{the quantity of raw oranges}) - (0.149 * \text{the quantity of juice})$ . This equation goes exactly

through observations 2, 3, and 5 while nonzero deviations exist for observations 1, 4, and 6. The dual to this problem requires that the shadow prices be between -1 and +1. The dual variables equal these extreme limits when the deviation variables are in the solution. This is shown by the shadow prices on the observations 1, 4, and 6.

### 9.1.1.2 Comments

The minimization of total absolute deviations formulation has been used in three settings: the solution of regression problems (Charnes, Cooper and Ferguson; Klein; Fisher (1961); Wagner (1959); Wilson); goal programming problems (as covered in the multiple objective chapter), and risk analysis (as in the risk modeling chapter). The regression formulation is commonly used when non-normal errors are expected (see Wilson for discussion).

### 9.1.2 Minimization of Largest Absolute Deviation

Models can involve minimization of the largest absolute deviation rather than the sum (i.e., the maximum forecast error using the so-called Chebyshev criterion). Such a formulation would be expressed as in the equations

$$\begin{aligned} \text{Min } & \text{Max}_i \{ d_i^+, d_i^- \} \\ \text{s.t. } & d_i^+ = Y_i - \sum_j X_{ji} b_j \quad \text{for all } i \\ & d_i^-, b_j \leq 0 \quad \text{for all } i \text{ and } j \end{aligned}$$

where the variable  $d_i$  is the deviation under the  $i^{\text{th}}$  observation and  $b_j$  is the  $j^{\text{th}}$  parameter in the forecast equation. The other symbols are as defined in the previous section. The problem formulation is straight forward. Suppose that we define a variable  $d$  (without a subscript) which will equal the largest deviation and introduce two equations for each observation (I):

$$\begin{aligned} d & \geq Y_i - \sum_j X_{ji} b_j \\ d & \geq \sum_j X_{ji} b_j - Y_i \end{aligned}$$

These equations require  $d$  to be greater than or equal to the deviation and the negative of the deviation for each observation. Thus,  $d$  will be greater than or equal to the absolute deviation from each equation. Taking a simple example without  $b$  variables, with observations on  $Y$  equaling -3, 2, and 7, then these equations become

Observed Constraints

$Y_i$	, \$ $Y_i$	, \$ & $Y_i$
-3	, \$ & 3	, \$ 3
2	, \$ 2	, \$ & 2
7	, \$ 7	, \$ & 7

Clearly,  $b_0$  cannot be less than 7 (the largest absolute deviation in the model). Since the objective function minimizes  $b_0$ , subject to these two constraints for each observation, the model collectively minimizes the maximum absolute value. The composite linear program is:

$$\begin{aligned}
 & \text{Min } b_0 \\
 \text{s.t. } & -b_0 - \sum_j X_{ji} b_j \leq -Y_i \quad \text{for all } i \\
 & -b_0 + \sum_j X_{ji} b_j \leq Y_i \quad \text{for all } i \\
 & b_j \leq 0 \quad \text{for all } j
 \end{aligned}$$

9.1.2.1 Example

Utilizing the data from the previous example with the restrictions that the intercept term  $b_0$  is unrestricted in sign but that the parameter on  $b_1$  be non-positive while the parameter  $b_2$  is non-negative. The resultant formulation is

Rows	,	$b_0$	$b_1$	$b_2$		Minimize
Objective	1					
1 <sup>+</sup>	-1	-1	-8	-5	#	-10
1 <sup>-</sup>	-1	1	8	5	#	10
2 <sup>+</sup>	-1	-1	-9	-1	#	-5
2 <sup>-</sup>	-1	1	9	1	#	5
3 <sup>+</sup>	-1	-1	-10	-9	#	-4
3 <sup>-</sup>	-1	1	10	9	#	4
4 <sup>+</sup>	-1	-1	-13	-8	#	-2
4 <sup>-</sup>	-1	1	13	8	#	2
5 <sup>+</sup>	-1	-1	-15	-2	#	-6
5 <sup>-</sup>	-1	1	15	2	#	6
6 <sup>+</sup>	-1	-1	-17	-3	#	-9
6 <sup>-</sup>	-1	1	17	3	#	9
			1		#	0
				1	\$	0

where all variables are non-negative and the GAMS formulation is called LARGE. This problem solution yields an objective function value of 3.722 with a variable and constraint solution as shown in Table 9.3.

The solution shows the regression line of the price of oranges is equal to  $7.167 - 0.111$  times the quantity of oranges. The maximum absolute deviation is present at the first, fourth and sixth observations equalling 3.722.

### 9.1.2.2 Comments

The above formulation solves the Chebyshev criterion problem as discussed in Wagner. This model form results in shadow price sum equaling 1 due to the duality constraint imposed by the form of  $y_0$ , as observed in the solution. Such a criterion has not been applied widely, but Wilson and Wagner give references.

### 9.1.3 Optimizing a Fraction

Charnes and Cooper (1962) present a LP formulation involving optimization of a fraction. This formulation allows problems maximizing such things as the average rate of return. The problem is

$$\begin{aligned} \text{Max} \quad & \frac{C_0 + \sum_j C_j X_j}{d_0 + \sum_j d_j X_j} \\ \text{s.t.} \quad & \sum_j a_{ij} X_j \leq b_i \text{ for all } i \\ & X_j \geq 0 \text{ for all } j \end{aligned}$$

where the denominator is strictly positive

$$d_0 + \sum_j d_j X_j > 0$$

Note there are constants in both the numerator and denominator accounting for exogenous terms which are not a function of the decision variables.

Transformation into a linear program requires several manipulations and substitutions, resulting in an exact transformation of the problem. First, define a variable  $y_0$  which equals one over the denominator

$$y_0 = \left[ d_0 + \sum_j d_j X_j \right]^{-1}$$

or equivalently

$$y_0^{-1} = d_0 + \sum_j d_j X_j$$

Multiplying both sides of this relationship by  $y_0$  yields

$$d_0 y_0 + \sum_j d_j X_j y_0 = 1$$

The new variable  $y_0$  is substituted into the above formulation, with the above relationship imposed. The net result is:

$$\begin{aligned}
\text{Max } & C_0 y_0 + \sum_j C_j X_j y_0 \\
\text{s.t. } & \sum_j a_{ij} X_j y_0 / y_0 \leq b_i \text{ for all } i \\
& d_0 y_0 + \sum_j d_j X_j y_0 = 1 \\
& y_0, X_j \geq 0.
\end{aligned}$$

Note that each  $a_{ij}X_j$  term has been multiplied by  $y_0/y_0$  which is simply 1. This will be convenient later. Now we introduce a change of variables. Let us define a new variable,  $y_j$  equal to the old variable  $X_j$  times  $y_0$ .

$$y_j = X_j y_0$$

Substituting this into the formulation above yields

$$\begin{aligned}
\text{Max } & C_0 y_0 + \sum_j C_j y_j \\
\text{s.t. } & \sum_j a_{ij} y_j / y_0 \leq b_i \text{ for all } i \\
& d_0 y_0 + \sum_j d_j y_j = 1 \\
& y_0, y_j \geq 0 \text{ for all } j
\end{aligned}$$

This formulation is not a LP problem; the term  $y_j/y_0$  appears in the first constraint equation. However, given that  $y_0$  (i.e. the reciprocal of the denominator) is strictly positive we can multiply both sides of the equation through by it without altering the direction of inequality

$$\sum_j a_{ij} y_j \leq b_i y_0$$

In turn, rewriting the second equation yields the LP formulation

$$\begin{aligned}
\text{Max } & C_0 y_0 + \sum_j C_j y_j \\
\text{s.t. } & -b_i y_0 + \sum_j a_{ij} y_j \leq 0 \text{ for all } i \\
& d_0 y_0 + \sum_j d_j y_j = 1 \\
& y_0, y_j \geq 0 \text{ for all } j
\end{aligned}$$

which is an exact transformation of the original fractional program. Once this problem has been solved, the levels

of the original optimum decision variables are easily discovered by performing the reverse transformation that  $X_j$  equals  $y_j$  divided by  $y_0$

$$X_j = y_j / y_0.$$

The LP form includes a new variable with coefficients in the matrix which are the negative of the right hand sides times a new variable ( $-b_i y_0$ ). A constraint is also added requiring the constant term in the denominator times the new variable ( $d_0 y_0$ ) plus the denominator terms involving the transformed variables to equal 1. The transformed model uses the same  $a_{ij}$ 's as the original. Its right hand sides are all 0's except the one in the new constraint. The objective function does not have a denominator term and the objective function altered to include the numerator constant times the new variable  $y_0$ . Model selection yields the optimal  $y$ 's ( $y_0, y_1, \dots, y_n$ ). Subsequently, then we transform to obtain  $X$ .

### 9.1.3.1 Example

Suppose that it is desirable to solve the following problem.

$$\begin{array}{rll} \text{Max} & \frac{1.8 X_1 + 1.7 X_2}{10 + 4 X_1 + 4.1 X_2} \\ \text{s.t.} & 1.5 X_1 + X_2 \# 6 \\ & 3.0 X_1 + 4 X_2 \# 20 \\ & X_1, X_2 \ \$ 0 \end{array}$$

Then the transformed problem is

$$\begin{array}{rll} \text{Max} & 1.8 y_1 + 1.7 y_2 \\ \text{s.t.} & -6 y_0 + 1.5 y_1 + y_2 \# 0 \\ & -20 y_0 + 3.0 y_1 + 4 y_2 \# 0 \\ & 10 y_0 + 4.0 y_1 + 4.1 y_2 = 1 \\ & y_0, y_1, y_2, \ \$ 0 \end{array}$$

Once a solution to this problem is obtained, the values of the original variables are recovered using the formulas

$$X_1 = y_1 / y_0$$

$$X_2 = y_2 / y_0$$

The GAMS model is set up in the file FRACTION and the solution is shown in Table 9.4.

The solution shows that the reciprocal of the denominator equals .031513 and that the decision variables are .042 and .126. Transforming these variables to their original values by dividing them through by the denominator reciprocal yields  $X_1=1.333$  and  $X_2=4$ . Plugging back into the original problem, the numerator equals 9.2; the denominator, 31.73, and their fraction 0.29 (the objective function value reported). One may also recover the shadow prices. In this case since the rows are multiplied by one over the denominator, the original shadow prices may be recovered by multiplying through by the denominator as shown in the scaling discussion in Chapter 18 . Thus the effective shadow price for constraint 1 is 10.85, and constraint 2 is 1.33. Constraint 3 has no analogue in the original problem, and thus, the shadow prices are not transformed.

#### 9.1.3.2 Comments

This is an exact transformation as long as the denominator remains strictly positive. The formulation fails if  $y_0$  equals zero in the optimal solution.

Much research has been done on fractional programming. The original development appears in Charnes and Cooper (1962). A historical perspective and literature review can be found in Schaible and Ibaraki.

## **9.2 Approximations**

Approaches to nonlinear problems often utilize approximations. Such approximations may be either one-time or iterative. Discussion of the one-time approximations constitutes the majority of the material below.

### **9.2.1 Grid Point Approximations**

Virtually all one-time approximations use grid points, representing nonlinear phenomena as a discrete series of linearized steps. Such approximations have been used where: a) costs increase with production; b) prices decrease as sales increase; and c) production yields decrease as input usage increases. All these cases involve decreasing returns to scale (increasing returns to scale are covered in the integer programming chapters). Approximations for decreasing returns cases use a set of discrete grid points assuming that: production cost, output prices, and/or quantities produced are constant between grid points, but change as we move along the grid.

#### 9.2.1.2 Functions with Separable Variables

The most common grid point approximation is separable programming. Separable programming deals with problems in which the functions may be of any nonlinear form, but must be separable into functions of a single variable. For example in the two variable case the functions  $f(x,y)$  must be decomposable into  $h(x) + g(y)$ .

Separable programming is usually considered a nonlinear programming technique (Hadley, 1964); but is commonly used in an LP setting. The most commonly used form of separable programming arose originally with

Charnes and Lemke, and was extended by Miller. The formulation yields an LP whenever the objective function terms are concave and the feasible set is convex (Hadley, 1964, p. 124). When these properties do not hold, more general separable programming needs to be used.

Separable programming relies on a set of grid points and constructs an approximation between these points. The approximation is setup so that the approximated value equals the value at the base point plus the slope divided by the difference from the base point. Suppose we wish to approximate the function at point X which falls between approximating points  $\hat{X}_k$  and  $\hat{X}_{k+1}$ . This can be expressed algebraically by the formula

$$f(X) \approx F(X) = f(\hat{X}_k) + \frac{f(\hat{X}_{k+1}) - f(\hat{X}_k)}{\hat{X}_{k+1} - \hat{X}_k} (X - \hat{X}_k)$$

In this case, if we write X as a convex combination of  $\hat{X}_k$  and  $\hat{X}_{k+1}$

$$\begin{aligned} X &= \delta_k \hat{X}_k + \delta_{k+1} \hat{X}_{k+1} \\ \delta_k + \delta_{k+1} &= 1 \\ \delta_k, \delta_{k+1} &\geq 0 \end{aligned}$$

where the new variables  $\delta_k$  and  $\delta_{k+1}$  are the amount of the  $k^{\text{th}}$  and  $k+1^{\text{st}}$  approximation points used.

Substituting this relationship into the above equation for F(X) we get the equation

$$F(X) \approx \delta_k f(\hat{X}_k) + \delta_{k+1} f(\hat{X}_{k+1})$$

where the function value is approximated by a convex combination of the function evaluated at the two adjacent grid points. This can be represented by a LP problem. Namely given the separable nonlinear problem

$$\begin{aligned} \text{Max } & \sum_j f_j(X_j) \\ \text{s.t. } & \sum_j g_{ij}(X_j) \leq b_i, \text{ for all } i \\ & X_j \geq 0, \end{aligned}$$

we may form the approximating problem



$$\begin{aligned}
\text{Max} \quad & \sum_{j, \mu} \delta_{j\mu} f_j(\hat{X}_{j\mu}) \\
\text{s.t.} \quad & \sum_{j, \mu} \delta_{j\mu} g_{ij}(\hat{X}_{j\mu}) \leq b_i, \quad \text{for all } i \\
& \sum_{\mu} \delta_{j\mu} = 1, \quad \text{for all } j \\
& \delta_{j\mu} \geq 0, \quad \text{for all } j \text{ and } \mu
\end{aligned}$$

where  $\hat{X}_{j\mu}$  is the  $\mu^{\text{th}}$  approximating point for  $X_j$  and

$$X_j = \sum_{\mu} \delta_{j\mu} \hat{X}_{j\mu}$$

This formulation involves a change of variables. The variables  $\delta_{j\mu}$  give the amount of the  $\mu^{\text{th}}$  grid point used in the approximation of the  $j^{\text{th}}$  variable. The terms  $f_j(\hat{X}_{j\mu})$  and  $g_{ij}(\hat{X}_{j\mu})$  give the values of the objective function and constraint terms evaluated at the various grid points. The new constraints on the  $\delta$  variables cause a convex combination of the grid points to be chosen for each variable approximated. The functions must be properly behaved, otherwise the nonzero  $\delta$ 's in the solution will not necessarily be adjacent; and the approximation will not work properly (Hadley, 1964). That is why users of the approximation should be careful to ensure that diminishing returns to production are present whenever this approach is being used.

#### 9.2.1.1.1 Example 1.

Suppose we approximate the problem.

$$\begin{aligned}
\text{Max} \quad & (4 - .25 X) X - (1 + .25 Z) Z \\
& X - 3 Y \leq 0 \\
& 2 Y - Z \leq 0 \\
& X, Y, Z \geq 0
\end{aligned}$$

To set this problem up, suppose we use values of  $X$  equal to 1,2,3,4,5,6 and the same values for  $Z$ . The separable programming representation is

Obj	3.75 $\delta_1$	7 $\delta_2$	9.75 $\delta_3$	12 $\delta_4$	13.75 $\delta_5$	15 $\delta_6$	&	1.25 $\delta_1$	&	3 $\delta_2$	&	5.25 $\delta_3$	&	8 $\delta_4$	&	11.25 $\delta_5$	&	15 $\delta_6$	max			
Output	$\delta_1$	$\delta_2$	$\delta_3$	$\delta_4$	$\delta_5$	$\delta_6$	&	3Y											#	0		
Input							&	2Y	&	$\delta_1$	&	2 $\delta_2$	&	3 $\delta_3$	&	4 $\delta_4$	&	5 $\delta_5$	&	6 $\delta_6$	#	0
Convex $\delta$	$\delta_1$	$\delta_2$	$\delta_3$	$\delta_4$	$\delta_5$	$\delta_6$														'	1	
Convex \$								$\delta_1$	%	$\delta_2$	%	$\delta_3$	%	$\delta_4$	%	$\delta_5$	%	$\delta_6$		'	1	
Nonneg	$\delta_1$	$\delta_2$	$\delta_3$	$\delta_4$	$\delta_5$	$\delta_6$	Y,	$\delta_1$	,	$\delta_2$	,	$\delta_3$	,	$\delta_4$	,	$\delta_5$	,	$\delta_6$	,	\$	0	

Note that  $\delta_2$  stands for the amount of the gridpoint  $X=2$  utilized having an objective value equal to the nonlinear function of  $X$  evaluated at  $X=2$ . The GAMS formulation is called SEPARABL and the resultant solution is shown in Table 9.5. The objective function value is 7.625. The model sets  $\delta_4=\delta_5=0.5$  amounting to 50% of gridpoint  $X_4$  and 50% of  $X_5$  or  $X=4.5$ . The value of  $Y = 1.5$ . Simultaneously  $\$1 = 1$  implying  $Z = 3$ . Now, let us examine the adequacy of the approximation. The objective function approximation for  $X$  has  $12(.5) + 13.75(.5) = 12.875$ , while the true  $f(X) = 12.9375$ . The  $Z$  approximation has zero error in this case. The modeler could either accept this as an adequate approximation or enter new grid points in the neighborhood of this solution.

#### 9.2.1.1.2 Example 2: Separable Terms in the Constraints

The above example deals with the approximation of separable objective function terms which McCarl and Onal found computationally unattractive. On the other hand, separable programming can also approximate constraint nonlinearities, which McCarl and Onal found attractive.

Suppose we wish to approximate the following problem

$$\begin{aligned} \text{Max} \quad & 3X - 3Y \\ \text{s.t.} \quad & X - (20 + 2Y + .2Y^2) \leq 0 \\ & X, Y \geq 0 \end{aligned}$$

Selecting a grid for  $Y$  of 0, 1, 2, 3, 4 and 5, the separable programming formulation becomes

<i>Obj</i>	$3X - 0\delta_1 - 3\delta_2 - 6\delta_3 - 9\delta_4 - 12\delta_5 - 15\delta_6$	max
<i>x bal</i>	$X - 20\delta_1 - 21.8\delta_2 - 23.2\delta_3 - 24.2\delta_4 - 24.8\delta_5 - 25\delta_6$	$\leq 0$
<i>convex</i>	$\delta_1 + \delta_2 + \delta_3 + \delta_4 + \delta_5 + \delta_6$	$= 1$
<i>nonneg</i>	$X, \delta_1, \delta_2, \delta_3, \delta_4, \delta_5, \delta_6$	$\geq 0$

The resultant GAMS model is in the file CONSEPAR and the solution is given in Table 9.6. We may

plug this solution back into the original problem to evaluate the adequacy of the approximation. The values of 8 imply that Y equals 3. However, optimization using calculus shows the optimum to be at Y equals 2.5, giving a yield of 23.75 and profits of 63.75. Thus, this demonstrates a 0.235 percent error of approximation. Again, one could go on to add more grid points, or accept the approximation error.

### 9.2.1.2 Gridpoints and Gridpoint Refinements

The separable formulation uses gridpoints to approximate functions. Readers may wonder how to define such points. Gridpoints are always defined in the context of the applied problem. The gridpoints should provide a reasonable approximation of the function in the domain of the answer, including points both close to the expected answer as well as points depicting functional extremes (Geoffrion (1977) discusses the importance of the extreme points). Even spacing of the gridpoints is not required. Güder and Morris show minimum theoretical error occurs with equal spacing. Thus, one could approximate a curve at the points 10, 2, 1, .95, .50, .10, .02 and .01. The gridpoint also may be redefined given a solution where, for example, one might find a solution of X = 2.50, discover the approximation is inadequate at that point, and then enter more gridpoints in the neighborhood of 2.5. Gridpoint refinement schemes are discussed in Bazaraa and Shetty. Implementation of a gridpoint refinement scheme is discussed in Kochenberger, Woolsey and McCarl.

### 9.2.1.3 Gridpoint Approximation of Functions of Multiple Variables

Gridpoint approximation may also be applied to functions containing multiple variables. In this case a multi dimensional grid is defined. This approach generally only works when one is approximating functions that depict a concave objective function and a convex constraint set.<sup>6</sup> The method involves techniques similar to separable programming and was developed by Dantzig and Wolfe. This method is discussed in Duloy and Norton; Shapiro (1979b); Bradley, Hax and Magnanti; and Lasdon. One of the possible formulations on this scheme which can be used is

$$\begin{array}{ll}
 \text{Max} & CX - \sum_j d_j Y_j \\
 \text{s.t.} & X - H(Y_1, Y_2, Y_3, \dots, Y_n) = 0 \\
 & Y_j \leq b_j \text{ for all } j \\
 & X, Y \geq 0
 \end{array}$$

---

<sup>6</sup> Readers unfamiliar with concavity and convexity should look at the Non-Linear Programming Theory chapter.

where there are multiple inputs and one output (for simplicity). The output  $X$  is a function of the levels of the multiple inputs ( $Y_j$ ). Also the function  $H(Y_1 \dots Y_n)$  has to be such that this problem has a convex constraint set.

We will discuss two versions of this formulation. The first version deals with cases where  $H$  is homogeneous of degree one and the second where  $H$  is homogeneous of degree less than one.

### 9.2.1.3.1 Homogeneous of Degree 1

The function  $H$  being homogeneous of degree 1 implies that

$$H(\lambda Y) = \lambda H(Y)$$

Suppose we choose a set of rays  $\hat{Y}_{ju}$  which depict the way each  $Y_j$  participates in each ray and define the variable  $\lambda_u$  which is the amount of ray  $\hat{Y}_{ju}$  which is used. Then we know that

$$H(\lambda_u \hat{Y}_{ju}) = \lambda_u H(\hat{Y}_{ju})$$

i.e., the function  $\lambda$  times the ray values equal  $\lambda$  times the function evaluated at the base ray point. The generalized programming formulation then becomes

$$\begin{aligned} \text{Max } & C X & - & \sum_j d_j Y_j \\ \text{s.t. } & X - \sum_u \lambda_u H(\hat{Y}_{ju}) & = & 0 \\ & \sum_u \lambda_u Y_{ju} - Y_j & = & 0 \quad \text{for all } j \\ & & & Y_j \leq b_j \quad \text{for all } j \\ & X, \lambda_u, & & Y_j \geq 0 \quad \text{for all } u \text{ and } j. \end{aligned}$$

The approximating model has the rays represented by a variable indicating how much of a particular ray  $Y_{ju}$  combination is used. They should be defined with unique ratios of the variables within  $\hat{Y}_{ju}$  (i.e., 1:1, 1:4, 4:1, etc., as below).

#### Example

This formulation is probably best illustrated by example. Consider the problem

$$\begin{aligned} \text{Max } & 4X - 20Y_1 - 100Y_2 \\ \text{s.t. } & X - 21Y_1^{0.75} Y_2^{0.25} = 0 \\ & Y_1 \leq 50 \\ & X, Y_1, Y_2 \geq 0. \end{aligned}$$

Suppose we select a set of combinations for  $Y_1$  and  $Y_2$ , given in Table 9.7, showing that when 1 unit of  $Y_1$  and 4 units of  $Y_2$  are used,  $X=29.7$ . Similarly, when 8 units of  $Y_1$  and 8 units of  $Y_2$  are used,  $X=168$ .

The resultant formulation is

$$\begin{array}{rcl}
 \text{Max} & 4X & - 20Y_1 - 100Y_2 \\
 \text{s.t.} & X - 29.7x_1 - 168x_2 - 59.4x_3 & \leq 0 \\
 & x_1 + 8x_2 + 4x_3 - Y_1 & = 0 \\
 & 4x_1 + 8x_2 + x_3 - Y_2 & = 0 \\
 & Y_1 & \leq 50 \\
 & X, x_1, x_2, x_3, Y_1, Y_2 & \geq 0
 \end{array}$$

An isoquant graph of this situation is portrayed in Figure 9.1.

Note that the three lines in the graph stand for the combinations 4 to 1, 1 to 1 and 1 to 4. The connected line in the graph is the isoquant for output equals 168, and the linear segments show how the production process is represented.

The GAMS formulation of the problem is called HOMOGEN and the solution is given in Table 9.8. This solution implies input use in the ratio 4:1. We may wish to put more rays in the neighborhood of 4:1 or we may be willing to accept the approximation error.

### 9.2.1.3.2 Homogeneous of Degree Less Than One

Now we turn to the case where we do not have homogeneity of degree one. In this case, the function evaluated at  $\alpha$  times the vector of inputs  $Y$ , is less than  $\alpha$  times that functional value evaluated at one unit at  $Y$  providing  $\alpha$  is less than or equal to one.

$$H(\alpha Y) < \alpha H(Y).$$

Consider the multiplicative function

$$X = aY_1^{b_1} Y_2^{b_2} \dots Y_n^{b_n}, \quad a, b_j > 0.$$

We may set up a vector representation

$$Y_j = \sum_u Y_{ju}.$$

Under this substitution the function becomes

$$X = a \left( \sum_j Y_{ju}^{b_j} \right) \sum_u \dots$$

But, the sum of the exponents on  $\sum_u$  is less than one

$$\sum_j b_j < 1.$$

Thus, as one moves  $\alpha$  units along the ray the function only increases by a factor of  $\alpha$  to the  $b_j$  which results in an increase less than  $\alpha$  X. This problem exhibits diminishing returns to scale because as  $\alpha_u$  is increased, less and less output is produced per unit increase in  $\alpha_u$ . This particular problem, however, may be formulated as a linear problem. This approach has been called the "column generation method" as discussed in Shapiro (1979b). Specifically, suppose we choose grid points  $Y_{ju}$  and a set of a priori multipliers  $\alpha_{uL}$ . The problem then becomes

$$\begin{aligned}
 \text{Max } & CX - \sum_j d_j Y_j \\
 \text{s.t. } & X - \sum_{uL} H(Y_{ju}, \alpha_{uL}) \alpha_{uL} = 0 \\
 & \sum_{uL} (Y_{ju}, \alpha_{uL}) \alpha_{uL} - Y_j = 0 \text{ for all } j \\
 & \sum_{uL} \alpha_{uL} = 1 \\
 & Y_j \leq b_j \text{ for all } j \\
 & X, \alpha_{uL}, Y_j \geq 0
 \end{aligned}$$

The variables are  $\alpha_{uL}$  where u identifies the input combination and L the length along that input combination. The parameter  $\alpha_{uL}$  gives how far along the  $u^{\text{th}}$  ray we move. The sum of the  $\alpha$  variables are then equal to one. This is a combination of the separable programming and homogeneity of degree one formulations above.

### Example

Consider the example problem

$$\begin{aligned}
 \text{Max } & 0.5X - 2Y_1 - 2Y_2 \\
 & X - 21Y_1^{.5} Y_2^{.25} = 0 \\
 & Y_1 \leq 10 \\
 & Y_1, Y_2 \geq 0
 \end{aligned}$$

where the exponents sum to 0.75 so the function is homogeneous of degree less than one. If we then put in three different approximation rays 1 to 1, 1 to 2 and 2 to 1 in these cases, the resultant values of X are

$$X = 21^{\alpha_1 .75}, \quad X = 29.7^{\alpha_2 .75}, \quad X = 25^{\alpha_3 .75}$$

We obtain a function that along the lines  $X$  is equal to some constant times  $\alpha^{0.75}$ . We then develop a table of approximations (Table 9.9). The resultant formulation is in Table 9.10 and in the file NONHOMOG. Note, here we have four combinations for each ratio of inputs, each representing different multiples of  $\alpha_{uL}$ . The convexity constraint is needed to insure that the model uses no more than one unit of the first step and rather is forced to go into the latter steps of the production process. The solution of the problem is given in Table 9.11 and shows that the 4<sup>th</sup> step of the second ray is used resulting in the value for the variables of  $X=99.3$ ,  $Y_1= 10$ ,  $Y_2= 5$  with the objective function equal to 19.65.

### 9.2.1.3.3 Comments

We get many classroom questions as to why we have presented the above generalized approximation formulations. There are two reasons. First, they constitute an approximation that can be used when representing a relationship between multiple inputs and outputs (i.e., see Onal et al.). Such a case occurs in agricultural models when approximating fertilizer response functions containing two or more fertilizer inputs or when intercropping is modeled. Approximations have also involved more complex production functions, where the output is a function of multiple inputs. Second, following Dorfman (1953), this can be used as a conceptual model. Often modelers include a number of activities for the production of a good where the input combinations and outputs arise from experiments or observed behavior. In this case, one is representing the underlying production process without ever estimating it. Such a procedure is utilized in Erhabor and Job.

A second question involves the manner in which grid points are chosen. Again, as in the separable programming case, this is done in accordance with the problem. For example, when one knows common levels of input use, one might construct several combinations of deviations from these numbers in small increments. Thus, when fertilizer and herbicide are used commonly in the ratio 50 lbs. fertilizer to 1 gallon herbicide one might add 5 activities involving: 1) 1 gal. herbicide with 50 lbs. of fertilizer; 2) 1 gal. herbicide with 47.5 lbs. of fertilizer; 3) 1 gal. herbicide with 45 lbs. of fertilizer; 4) 1 gal. herbicide with 52.5 lbs. of fertilizer and 5) 1 gal. herbicide with 55 lbs. of fertilizer. In turn, the user should examine the model solution and see if the solutions chosen use the most extreme ray for an input (e.g., the least amount of herbicide possible per unit of fertilizer). In such a case one should consider entering alternatives expanding the space represented. The representation is only satisfactory when the solution is interior to the cone of approximation points used and not on its boundary (Shapiro and Geoffrion provide theoretical

explorations of related topics).

### 9.2.2 Iterative Approximations

In addition to the step approximation formulations above, there are a number of iterative approximations which can be used. We will not cover these in depth; they are largely numerical techniques. Those interested in alternative approaches should examine Zangwill's convex simplex method, Dembo's geometric programming condensation method, or the methods reviewed in Himmelblau; Reklaitis et al; or Bazaraa and Shetty.

We will explain one technique for illustrative purposes. The iterative approximation presented here was developed by Griffith and Stewart and is based on the concept of a Taylor series expansion. This method solves the problem

$$\begin{aligned} \text{Max } & f(X) \\ & g(X) \# b \\ & L_j \# X_j \# G_j \end{aligned}$$

using a first order Taylor series expansion. A first order Taylor series expansion assumes that a functional value can be represented as a first order expansion of the function evaluated at a base point plus the derivative of that base point times the difference of X from the base point. The approximating problem then is given by

$$\begin{aligned} \text{Max } & f(X_0) + \frac{d}{dX} f(X_0) (X - X_0) \\ \text{s.t. } & g_i(X) = g_i(X_0) + \frac{d}{dX} g_i(X_0) (X - X_0) \# b_i \\ & L_j \# X_j \# G_j \end{aligned}$$

where given a base point  $X_0$  we approximate the value at any X using a LP formulation to find the difference from  $X_0$  that the solution will move where all of the terms involving  $X_0$  are constants. This is done by substituting in a variable  $\mu_j$  such that

$$\mu_j = X_j - X_{0j}$$

to obtain the LP problem



$$\begin{aligned} \text{Max } f(X_0) \quad & \mu_j \frac{d}{dX_j} f(X_0) \\ & \mu_j \frac{d}{dX_j} g(X_0) \mu_j \quad \# \quad b_i - g_i(X_0) \\ & \& \text{Lim}_j^{\&} \quad \# \quad \mu_j \quad \# \quad \text{Lim}_j^{\%} \end{aligned}$$

where the limits are developed relative to an exogenous parameter  $\$j$

$$\text{Lim}_j^{\&} = \min[\$j, X_{0j} - L_j]$$

$$\text{Lim}_j^{\%} = \min[\$j, G_j - X_{0j}]$$

Here the variables are given by

$$X_{0j}^{k+1} = X_{0j}^k + \mu_j$$

Then, given any initial choice of variables at the  $k^{\text{th}}$  iteration, the variable at  $k^{\text{th}+1}$  iteration is equal to that variable at the  $k^{\text{th}}$  iteration plus the optimal change variable value  $\mu_j$ . The change variables are artificially constrained to be limited by some quantity  $\$j$ . It is desirable that this quantity becomes smaller as iterations proceed.

### 9.2.3 Other Approximations

We have covered only a few of the approximations which are possible in the area of nonlinear programming. There are also other approximations based on exotic transformations for various sorts of problems; e.g., see Dembo; or McCarl and Tice. Many approximations may be used given special problem structures. Their use depends on the ingenuity of the modeler. What we have attempted to do above is give some of the basic techniques and references.

## References

- Bazaraa, M.S. and C.M. Shetty. Nonlinear Programming: Theory and Algorithms. John Wiley and Sons, 1979.
- Bradley, S.P., A.C. Hax, and T.L. Magnanti. Applied Mathematical Programming. Menlo Park, CA: Addison-Wesley Publishing Company, Inc. 1977.
- Charnes, A. and C. Lemke. "Minimization of Nonlinear Separable Convex Functionals." Naval Research Logistics Quarterly. 1(1954):301-312.
- Charnes, A. and W.W. Cooper. "Programming with Linear Fractionals." Naval Research Logistics Quarterly. 9(1962):181-186.
- Charnes, A., W.W. Cooper and R.O. Ferguson. "Optimal Estimation of Executive Compensation by Linear Programming." Management Science. 1(1955):138-151.
- Dantzig, G.B. and P. Wolfe. "The Decomposition Algorithm for Linear Programs." Econometrica. 29(1961):767-778.
- Dembo, R.S. "The Current State-of-the-Art of Algorithms and Computer Software for Geometric Programming." Journal of Optimization Theory and Applications. 26(1978):149.
- Dorfman, R. "Mathematical or Linear Programming: A Nonmathematical Exposition." American Economic Review. 43(1953):797-825.
- Duloy, J.H. and R.D. Norton. "Prices and Incomes in Linear Programming Models." American Journal of Agricultural Economics. 57(1975):593-600.
- Erhabor, P.O. "Efficiency of Resource Use Under Small-Scale Irrigation Technology in Nigeria." Purdue University Water Resources Research Center Technical Report No. 148, 1982.
- Fisher, W.D. "A Note on Curve Fitting with Minimum Deviations by Linear Programming." Journal of the American Statistical Association. 56(1961):359-362.
- Geoffrion, A.M. "Objective Function Approximations in Mathematical Programming." Mathematical Programming. 13(1977):23-37.
- Griffith, R.E. and R.A. Stewart. "A Nonlinear Programming Technique for the Optimization of Continuous Processing Systems." Management Science. 7(1961):379-392.
- Güder, F. and J.G. Morris. "Objective Function Approximation: An Application to Spatial Price Equilibrium Models." American Journal of Agricultural Economics. 70(1988):391-396.
- Hadley, G. Nonlinear and Dynamic Programming. Reading, MA: Addison-Wesley Publishing Company, Inc., 1964.
- Himmelblau, D.M. Applied Nonlinear Programming. New York: McGraw-Hill, 1972.
- Job, M.O. "A Study of Technology Adoption by Subsistence Farmers With a Case Study Involving Dairy Goats in Kenya." Ph.D. Thesis, Purdue University, 1982.
- Klein, K.K. "Minimizing Absolute Versus Squared Deviations for Predicting Western Regional and National Hog Supplied in Canada." Canadian Journal of Agricultural Economics. 27(1978):74-84.
- Kochenberger, G.A., R. Woolsey, and B. McCarl. "On the Solution of Geometric Programs Via Separable Programming." Operational Research Quarterly. 24(1973):285-94.
- Lasdon, L.S. Optimization Theory for Large Systems. New York: Macmillan. 1970.

- McCarl, B.A. and H. Onal. "Linear Approximation Using MOTAD and Separable Programming: Should it be Done?" American Journal of Agricultural Economics. 71(1989):158-166.
- McCarl, B.A. and T. Tice. "Should Quadratic Programming Problems be Approximated?" American Journal of Agricultural Economics. 64(1982):585-89.
- Miller, C.E. "The Simplex Method for Local Separable Programming" in Recent Advances in Mathematical Programming, edited by R.L. Graves and P. Wolfe, 1963.
- Onal, H., B.A. McCarl, W.L. Griffin, G. Matlock, and J. Clark. "A Bioeconomic Analysis of the Texas Shrimp Fishery and Its Optimal Management." American Journal of Agricultural Economics. 73(1991):1161-1170.
- Reklaitis, G.V., A. Ravindran, and K.M. Ragsdell. Engineering Optimization: Methods and Applications. New York:John Wiley and Sons, 1983.
- Schaible, S. and T. Ibaraki. "Fractional Programming." European Journal of Operational Research. 12(1983):325-338.
- Shapiro, J.F. Mathematical Programming: Structures and Algorithms. New York: Wiley and Sons, 1979.
- Wagner, H.M. "Linear Programming and Regression Analysis." Journal of the American Statistical Association. 54(1959):206-212.
- Wilson, H. "Least Squares Versus Minimum Absolute Deviations Estimation in Linear Models." Decision Sciences. 9(1978):322-335.
- Zangwill, W.I. Nonlinear Programming: A Unified Approach. Englewood Cliffs, NJ: Prentice-Hall, 1969.

**Table 9.1. Minimization of Sum of Absolute Deviations Formulation**

	$x_1^+$	$x_1^-$	$x_2^+$	$x_2^-$	$x_3^+$	$x_3^-$	$x_4^+$	$x_4^-$	$x_5^+$	$x_5^-$	$x_6^+$	$x_6^-$	$b_0$	$b_1$	$b_2$	
Obj	1	1	1	1	1	1	1	1	1	1	1	1				Min
1	1	-1											1	8	5	=10
2			1	-1									1	9	1	=5
3					1	-1							1	10	9	=4
4							1	-1					1	13	8	=2
5									1	-1			1	15	2	=6
6											1	-1	1	17	3	=9

**Table 9.2. Solution of Minimization of Absolute Deviation Sum Example**

Objective function = 11.277					
Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
$s_1^+$	5.787	0	Obs 1	0	1
$s_1^-$	0	2.000	Obs 2	0	-0.660
$s_2^+$	0	1.66	Obs 3	0	0.191
$s_2^-$	0	0.340	Obs 4	0	-1
$s_3^+$	0	0.809	Obs 5	0	-0.532
$s_3^-$	0	1.191	Obs 6	0	1
$s_4^+$	0	2.000			
$s_4^-$	2.723	0			
$s_5^+$	0	1.532			
$s_5^-$	0	0.468			
$s_6^+$	2.766	0			
$s_6^-$	0	2.000			
$b_0$	3.426	0			
$b_1$	0.191	0			
$b_2$	-0.149	0			

**Table 9.3. Solution of Largest Absolute Deviation Example**

Variables	Value	Reduced Cost	Equation	Slack	Shadow Price
,	3.722	0	1 <sup>+</sup>	0	-0.222
b <sub>0</sub>	7.167	0	1 <sup>-</sup>	7.44	0.0
b <sub>1</sub>	-0.111	0	2 <sup>+</sup>	4.89	0.0
b <sub>2</sub>	0.000	2.056	2 <sup>-</sup>	2.56	0.0
			3 <sup>+</sup>	5.78	0.0
			3 <sup>-</sup>	1.67	0.0
			4 <sup>+</sup>	7.44	0.0
			4 <sup>-</sup>	0	-0.5
			5 <sup>+</sup>	3.22	0.0
			5 <sup>-</sup>	4.22	0.0
			6 <sup>+</sup>	0	-0.278
			6 <sup>-</sup>	7.44	0.0

**Table 9.4. Solution to the Example for Optimizing a Fraction**

Objective function = 0.2899					
Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
y <sub>0</sub>	0.032	0	1	0	0.342
y <sub>1</sub>	0.042	0	2	0	0.042
y <sub>2</sub>	0.126	0	3	0	0.290

**Table 9.5. Solution to the Step Approximation Example**

---

Objective function = 7.625

---

Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
$8_1$	0	-3.000	1	0	1.750
$8_2$	0	-1.500	2	0	2.625
$8_3$	0	-0.500	3	0	5.000
$8_4$	0.5	0	4	0	2.625
$8_5$	0.5	0			
$8_6$	0	-0.500			
Y	1.5	0			
$\$1$	0	-1.250			
$\$2$	0	-0.375			
$\$3$	1	0			
$\$4$	0	-0.125			
$\$5$	0	-0.750			
$\$6$	0	-1.875			

---

**Table 9.6. Solution to the Constraint Step Approximation Problem**

Objective function = 63.6

Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
X	23.2	0	1	0	3
$8_1$	0	-3.6	2	0	63.6
$8_2$	0	-1.2			
$8_3$	1	0			
$8_4$	0	0			
$8_5$	0	-1.2			
$8_6$	0	-3.6			

**Table 9.7. Set of  $Y_1, Y_2$  Combinations for Homogeneous of Degree 1 Example**

X	$Y_1$	$Y_2$
29.7	1	4
168	8	8
59.4	4	1

**Table 9.8. Solution to Example Problem for Homogeneous of Degree 1**

Objective function = 719.8

Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
X	742.5	0	1	0	4
" <sub>1</sub>	0	-315.6	2	0	34.4
" <sub>2</sub>	0	-403.2	3	0	100
" <sub>3</sub>	12.5	0			
$Y_1$	50	14.4			
$Y_2$	12.5	0			



**Table 9.9. Approximations for the Homogenous of Degree Less Than One Example**

X	Y <sub>11</sub>	Y <sub>12</sub>	X	Y <sub>12</sub>	Y <sub>22</sub>	X	Y <sub>13</sub>	Y <sub>23</sub>
21	1	1	29.7	2	1	25.0	1	2
59.4	4	4	49.9	4	2	70.6	4	8
80.5	6	6	67.7	6	3	95.7	6	12
118.1	10	10	99.3	10	5	140.4	10	20

**Table 9.10. Formulation of the Homogeneous Degree Less than One Example**

Rows	X	$\delta_{11}$	$\delta_{12}$	$\delta_{13}$	$\delta_{14}$	$\delta_{21}$	$\delta_{22}$	$\delta_{23}$	$\delta_{24}$	$\delta_{31}$	$\delta_{32}$	$\delta_{33}$	$\delta_{34}$	Y <sub>1</sub>	Y <sub>2</sub>	RHS
Obj	0.5													-2	-2	max
x bal	1	-21	-59.4	-80.5	-118.1	-29.7	-49.9	-67.7	-99.3	-25.0	-70.6	-95.7	-140.4			= 0
Y bal		1	4	6	10	2	4	6	10	1	4	6	10	-1		= 0
convex		1	4	6	10	1	2	3	5	2	8	12	20		-1	= 0
Y lim						1	1	1	1	1	1	1	1			# 1
														1		# 10

**Table 9.11. Solution to the Homogenous of Degree Less Than One Example**

---

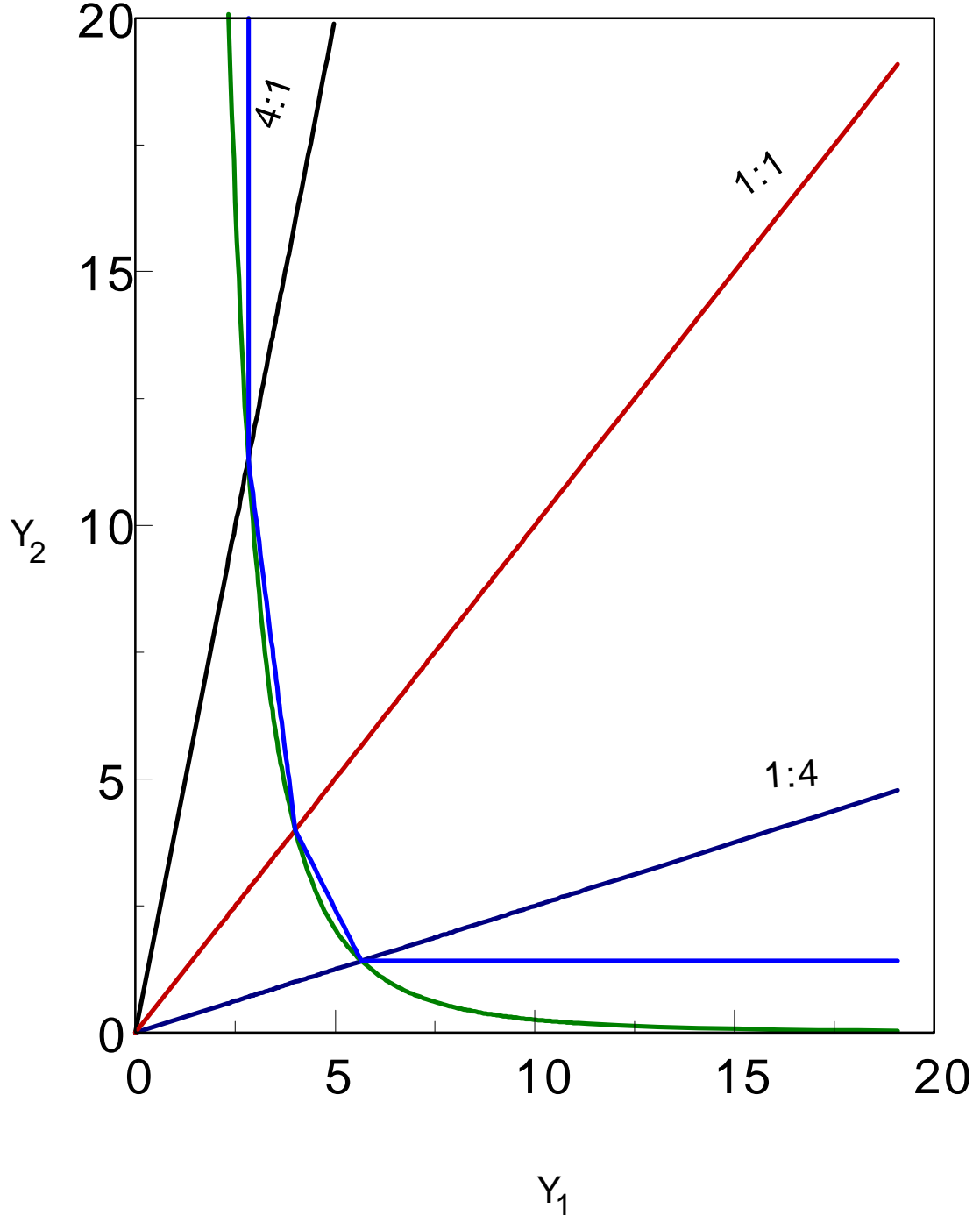
Objective function = 19.651

---

Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
X	99.3	0	x bal	0	0.500
$8_{11}$	0	-5.506	$Y_1$ bal	0	2.850
$8_{12}$	0	-0.856	$Y_2$ bal	0	2.000
$8_{13}$	0	0.000	convex	0	11.156
$8_{14}$	0	-0.606	Y lim	0	0.85
$8_{21}$	0	-4.006			
$8_{22}$	0	-1.581			
$8_{23}$	0	-0.404			
$8_{24}$	1	0.000			
$8_{31}$	0	-5.519			
$8_{32}$	0	-3.237			
$8_{33}$	0	-4.384			
$8_{34}$	0	-9.434			
$Y_1$	10	0.000			
$Y_2$	5	0.000			

---

Figure 9.1 Approximation of Homogeneous of Degree One Example



## CHAPTER X: MODELING SUMMARY

Now that LP theory and basic modeling have been covered, a number of additional considerations involved with using models are covered, including variable and constraint types as well as LP assumptions.

### 10.1 Types of Constraints and Variables in Linear Programming Models

In most text books the LP problem is vastly oversimplified when first defined. For example, consider the problem

$$\begin{array}{ll} \text{Max} & CX \\ \text{s.t.} & AX \# b \\ & X \geq 0 \end{array} .$$

Where, the X's are defined as alternative production processes while the constraints ( $AX \# b$ ) are referred to as resource limitations. However, the previous chapters show there may be many different types of variables and constraints within such a formulation. This section develops a characterization of the various possible types of variables and constraints which can be used.

#### 10.1.1 Types of Constraints

Possible constraint types include resource limitations, minimum requirements, supply-demand balances, ratio controls, upper/lower bounds, accounting relations, deviation constraints, and approximation or convexity constraints.

##### 10.1.1.1 Resource Limitations

Resource limitations depict relationships between endogenous resource usage and exogenous resource endowments. A resource limitation restricts endogenous resource use to be less than or equal to an exogenous resource endowment. An example of a resource limitation constraint is

$$3X_1 + 4X_2 \leq 7$$

This constraint requires the sum of resources used in producing  $X_1$ , which uses 3 resource units per unit, plus those used in producing  $X_2$ , which uses 4 resource units per unit, to be no greater than an exogenous resource endowment of 7 units. Resource usage depends on the values of  $X_1$  and  $X_2$  determined by the model and thus is an endogenous quantity. This type of constraint appears in many of the formulations in Chapter 5, including the resource allocation problem.

### 10.1.1.2 Minimum Requirements

Minimum requirement constraints require an endogenously determined quantity to be greater than or equal to an exogenously specified value. A simple illustration is

$$X_1 + 2X_2 \geq 4$$

In this case the endogenous sum of  $X_1$  plus two times  $X_2$  is constrained to be greater than or equal to the exogenously imposed requirement of four. One may also express this constraint in less-than-or-equal-to form as

$$-X_1 - 2X_2 \leq -4$$

The minimum requirement often specifies that the model must meet exogenous demand through the endogenous supply of goods. This kind of constraint is present in many different types of programming models. An example appears in the transportation model of Chapter 5.

### 10.1.1.3 Supply and Demand Balance

The supply-demand balance requires that endogenous supply be balanced with endogenous demand. A typical example is

$$X_1 \leq X_2$$

This equation requires the endogenous demand for a good ( $X_1$ ) to be less than or equal to the endogenous supply of that good ( $X_2$ ). After moving all the variables to the left hand side, the constraint becomes

$$X_1 - X_2 \leq 0.$$

More generally, supply demand balances may involve exogenous quantities. Consider the inequality

$$2X_1 - X_2 \leq 3.$$

Here, the difference between endogenous demand ( $2X_1$ ) and supply ( $X_2$ ) is less than or equal to an exogenous supply of 3 units. This inequality can also be expressed in the following form:

$$2X_1 - X_2 \leq 3$$

which says that the endogenous demand ( $2X_1$ ) must be less than or equal to total supply, which consists of endogenous supply ( $X_2$ ) plus exogenous supply (3). A related situation occurs under the constraint

$$X_1 - 4X_2 \leq -2.$$

Here, the difference between endogenous supply and endogenous demand is less than or equal to minus 2.

This can be rewritten as

$$X_1 - 2 \leq 4X_2$$

which states that endogenous demand ( $X_1$ ) plus exogenous demand (2) is less than or equal to endogenous supply ( $4X_2$ ). In general, supply-demand balances are used to relate endogenous supply and demand to

exogenous supply and demand. The general case is given by

$$\text{Demand}_{\text{En}} + \text{Demand}_{\text{Ex}} \leq \text{Supply}_{\text{En}} + \text{Supply}_{\text{Ex}} .$$

Here, the sum of demand over endogenous and exogenous sources (respectively denoted by the subscripts En and Ex) must be less than or equal to the supply from endogenous and exogenous sources.

Manipulating the endogenous variables to the left hand side and the exogenous items to the right hand side gives

$$\text{Demand}_{\text{En}} - \text{Supply}_{\text{En}} \leq \text{Supply}_{\text{Ex}} - \text{Demand}_{\text{Ex}} .$$

Here endogenous demand minus endogenous supply is less than or equal to exogenous supply minus exogenous demand.

This constraint contains the resource limitation and minimum requirement constraints as special cases. The resource limitation constraint exhibits zero endogenous supply and exogenous demand. The minimum requirement constraint exhibits zero endogenous demand and exogenous supply.

Supply demand balances are present in many of the examples of Chapter 7. The assembly, disassembly, assembly - disassembly, and the sequencing problems all possess such constraints.

#### 10.1.1.4 Ratio Control

Ratio control constraints require the ratio of certain endogenous variables to be no more than an endogenous sum, possibly influenced by exogenous factors. Specifically suppose that a number of units of  $X_1$  have to be supplied with every unit of  $X_2$ . For example, a LP formulation of an automobile manufacturer might require a constraint to insure that there are four tires for every car sold. Such a situation would be modeled by

$$4X_2 \leq X_1 \leq 0$$

where  $X_1$  is the number of tires and  $X_2$  the number of cars sold. In order for one unit of  $X_2$  to be sold, 4 units of  $X_1$  must be supplied.

The general case is depicted by

$$\text{EN}_{\text{rat}} \leq p (w_{\text{EN}} \text{EN}_{\text{rat}} + \text{EN}_{\text{other}} + \text{EX}_{\text{other}}).$$

where the left hand side elements are denoted with the subscript "rat," and the right hand side elements with "other." EN denotes endogenous variables and EX denotes exogenous constants. The parameter  $w_{\text{EN}}$  is nonzero only when the endogenous variables ( $\text{EN}_{\text{rat}}$ ) are part of the right hand side. The constraint requires that the endogenous "rat" expression be less than or equal to p times the sum of the "rat" term or variables plus the "other." Manipulating this constraint so that all the endogenous variables are on the left hand side gives

$$(1 - p_{w_{EN}})EN_{rat} - p_{EN_{other}} \# p_{EX_{other}}$$

This expression is rather abstract and is perhaps best seen by the example. Suppose we wish the variable  $X_1$  to be no more than 25 percent of  $X_1 + X_2$ . Thus

$$X_1 \# .25 (X_1 + X_2)$$

Placing all the endogenous variables on the left hand side yields

$$.75X_1 - .25X_2 \# 0$$

Consider another example which includes exogenous factors. Suppose that

$$(X_1 + 3) \# .25 (2X_1 + 3X_2 + 10)$$

this can be written as

$$.50X_1 - .75X_2 \# -5$$

Here we have a requirement between  $X_1$  and  $X_2$  and an exogenous constant appearing on the right hand side. Finally, if the endogenous variables do not appear on the right hand side (for example, where  $X_1$  is less than or equal to one-third the sum of  $X_2 + 4X_3$ ) then the inequality would be manipulated to state:

$$EN_{Rat} - p_{EN_{oth}} \# 0$$

within the example context yields

$$X_1 - 1/3X_2 - 4/3X_3 \# 0$$

This is an example where the  $w$ 's in the ratio control constraint are zero.

This particular constraint type is a special case of the supply/demand balances. It is not used explicitly in any of the general formulations, but would also be used in a feed problem formulation where the quantity of feed to be produced was not exogenously given (i.e., on the right hand side) but rather was an endogenous variable.

#### 10.1.1.5 Bounds

Upper and lower bounds have important implications for the performance of the simplex algorithm. Upper bounds are resource limitation constraints; however, they only involve a single variable. Similarly, lower bounds are minimum requirement constraints on a single variable. Examples are

$$X_1 \# 4$$

$$X_1 \# 2.$$

Such constraints are usually exploited by LP solvers so that they do not enter the basis inverse.

#### 10.1.1.6 Accounting Relations

Accounting relations are used to add endogenous sums for model solution summary purposes. These relations are used for modeler convenience in summarizing a solution (i.e., adding up total labor utilized by crop). Accounting relations can be depicted as either

$$\sum_{j=1}^n A_{ij}X_j \leq 0 \quad \text{or} \quad \sum_{j=1}^n A_{ij}X_j + S = 0$$

In the first case the surplus variable would equal the sum of AX (assuming AX is always non-negative). The second form of the equation simply introduces an accounting variable which takes on the value of the sum. Accounting relations are discussed in the purposeful modeling section.

#### 10.1.1.7 Deviation Constraints

Deviation constraints are used to develop the endogenous deviation of a particular sum from a target level. The general format of these constraints is as follows:

$$\sum_j g_{ij}x_j \pm \text{Dev}_i = T_i$$

Here  $T_i$  is a target level and  $\text{Dev}_i$  is a deviation variable indicating the amount the endogenous sum ( $\sum g_{ij}x_j$ ) deviates (as measured by the deviation variable  $\text{Dev}_i$ ) from the target level ( $T_i$ ). The deviation constraint concept is utilized in the nonlinear transformations involving absolute value, multi-objective programming, and risk modeling.

#### 10.1.1.8 Approximation or Convexity Constraints

A convexity constraint requires the sum of a set of variables to be equal to or possibly less than or equal to one. These are commonly used in approximations such as those under the separable programming section of the nonlinear approximations chapter.

### **10.1.2 Types of Variables**

There are many different types of variables. Production, sales, purchase, transformation, slack, surplus, artificial, step, deviation and accounting variables are discussed in this section.

#### 10.1.2.1 Production Variables

Production variables depict the production of outputs from inputs. Such a variable is represented by

$X_2$  in the LP problem



$$\begin{array}{rcll}
(1) & \text{Max} & aX_1 & \\
(2) & \text{s.t.} & X_1 & \# \quad 0 \\
(3) & & cX_2 & \# \quad 0 \\
(4) & & eX_2 & \# \quad f \\
(5) & & hX_2 & \# \quad 0 \\
(6) & & jX_2 & \# \quad 0 \\
(7) & & kX_2 & \# \quad T \\
(8) & & pX_2 & \# \quad 0 \\
(9) & & rX_2 & \# \quad b \\
& & & X_1 \quad \$ \quad 0
\end{array}$$

Note that  $X_2$  produces items which are transferred into the equations (2) and (8). The  $X_2$  variable also uses inputs from equations (3) and (5) and utilizes a fixed resource which is represented by (4). Thus,  $X_2$  depicts a multi-factor/multi-product production process. Production coefficients do not always explicitly appear in the constraint equations; rather, production may simply yield revenue in the objective function as in the resource allocation and sequencing problems. Production activities may also use inputs which have pre-specified costs, thus the objective function coefficients may involve revenue and/or cost terms. The purposeful modeling section provides such an example.

#### 10.1.2.2 Sales Variables

Sales variables reflect the sale of an item at an exogenously determined price. For example, variable  $X_1$  in the above tableau depicts the sale of an item at price  $a$ , where the item is drawn from the supply-demand balance that relates  $X_1$  to the production activity  $X_2$  (equation (2)).  $X_9$  is also a sales variable. Sales variables appear in numerous examples above. For example, see the assembly-disassembly and joint product formulations.

#### 10.1.2.3 Purchase Variables

Purchase variables depict the purchase of items at exogenously specified prices with the items made available for use within the model. Examples of this type of variable are  $X_3$  and  $X_4$  above. For example, one unit of  $X_3$  yields one unit of supply to the supply-demand balance equation (3) and enters the objective function with a coefficient of  $-d$ . Purchase activities are illustrated in the assembly formulation.

#### 10.1.2.4 Transformation Variables

Transformation variables transform the location, time availability, unit or form characteristics of an

item (although other inputs may be required to do this). Examples of such variables include transportation variables which alter location, storage variables which alter time availability, unit transformation variables that convert the units from, say tons to pounds, or variables which transform a good from one form to another, possibly with the addition of other inputs. An example of this type of variable includes beef slaughter, where pounds of beef on the hoof are converted into hanging carcass beef.

The variable  $X_5$  in the LP example given by (1) - (9) is a transformation variable depicting transformation at per unit cost I of the resources in constraint (3) into the resources in constraint (4). Transformation variables appear in the storage and transportation examples.

#### 10.1.2.5 Slack Variables

Slack variables represent the amount of excess resources (i.e., resources which are unused in production). Ordinarily, they have a zero objective function coefficient and a plus one entry in a single constraint. Slack variables are defined in association with less than or equal to constraints representing the extent to which the endogenous quantity is less than the right hand side. Slack variables do not play a large role in model formulations (although deviation and accounting variables are forms of slack variables). However, slack variables can play an important role in solution interpretation. Modelers should check which resources are left unused (with non-zero slack) and question whether such a situation is reasonable.

#### 10.1.2.6 Surplus Variables

Surplus variables are analogous to slack variables; they have zero objective function coefficients and a coefficient only in one particular row. They represent the amount that the left hand side of a constraint is greater than the right hand side. Surplus variables do not ordinarily play a large role in applied modeling. However, they may be important in the interpretation of the solution of a model. For example, the magnitude of a surplus variable may indicate the extent to which over-production occurs above a minimum requirement.

#### 10.1.2.7 Artificial Variables

Artificial variables are most often utilized to make an infeasible problem feasible, allowing the violation of equality constraints or minimum requirements. Artificial variables ordinarily have a large cost in the objective function and a coefficient in the particular row with which they are associated. However, artificial variables can play a role in applied modeling. For example, artificial variables can be used to prohibit an infeasible solution from arising in solvers. Artificial variables also play an important role in discovering the causes of infeasibilities, as discussed in the chapter on debugging models.

#### 10.1.2.8 Step Variables

Linear programs may involve the approximation of nonlinear phenomena. Step variables are often used in such approximations. One may, for example, utilize step variables to represent different portions of an increasing cost function. Step variables receive their name from their portrayal of nonlinear functions as a series of piece-wise linear steps. Step variables appear in the separable programming formulations.

#### 10.1.2.9 Deviation Variables

Deviation variables tell the amount by which an endogenous sum deviates from a target value. Such variables are illustrated in the LP model given by (1)-(9) by  $X_7$  or  $X_8$ . For example, in equation (7), these variables indicate the amount  $kX_2$  deviates from the target value  $T$ . The variable  $X_7$  gives the amount that the sum is over the target while the variable  $X_8$  gives the amount the sum is under the target. These variables are analogous to surplus and slack variables; however, they may have an objective function coefficient which reflects costs or revenues associated from "missing" the target. These variables will work properly as long as the objective function is properly structured as explained in the multi-objective programming chapter. Deviation variables are also an important part in the LP approaches to regression (as used in the absolute value formulation) and in the MOTAD formulation.

#### 10.1.2.10 Accounting Variables

An accounting variable is typically used to indicate the value of endogenous sums so that the analyst need not manually summarize the solution. The variable  $X_6$  in equation (6) is an example of this type of variable. These variables are also prominently featured in the section on purposeful modeling.

### **10.2 "Violations" of the Algorithmic Assumptions**

The algorithmic assumptions of LP hold for individual variables within a linear program but not necessarily for the total process represented. Thus, modeling techniques can be used to generate formulations which, for practical purposes, invalidate the algorithmic assumptions. Let us consider models which nominally appear to violate each of the algorithmic assumptions.

#### **10.2.1 Nonproportional Example**

It is possible to satisfy the algorithmic assumptions regarding proportionality while formulating nonproportional problems. For example, suppose a production process exhibits diminishing returns to scale (i.e., doubling the level of inputs does not double the output). This may be modeled as follows:



### 10.2.2 Non-Additive Example

Models may also be constructed which appear to violate the additivity assumption. Suppose a production process involves two inputs which can be substituted in production. This may be modeled as follows:

$$\begin{array}{rcl}
 \text{Max} & 3Y & \\
 \text{s.t.} & Y & \leq 2X_1 + 2X_2 + 2X_3 \leq 0 \\
 & & 4X_1 + 2X_2 + X_3 \leq r_1 \\
 & & X_1 + 2X_2 + 4X_3 \leq r_2 \\
 & Y, & X_1, X_2, X_3 \geq 0.
 \end{array}$$

Note this formulation depicts the production of Y using production processes  $X_1$ ,  $X_2$ , or  $X_3$ . Each process produces 2 units of Y; however, inputs are used in different proportions.  $X_1$  uses four units of input 1 and one unit of input 2;  $X_2$  utilizes equal combinations of the two inputs, while  $X_3$  uses one unit of input 1 with four units of input 2. The formulation is constrained by input availability where the quantity inputs available are designated as  $r_1$  and  $r_2$ .

Now let us illustrate the nonadditive nature of this formulation. Suppose equal amounts of the inputs are available ( $r_1 = r_2$ ), then it would be optimal to produce in a pattern utilizing the inputs in equal proportions. Note that by producing  $X_1$  and  $X_3$  in equal amounts, the inputs would be used in equal proportion, i.e., setting both variables to one would produce 1.6 units of output while utilizing 2 units of inputs of  $r_1$  and  $r_2$ . Thus, 1.6 units of output are attained when using 0.4 units of each variable. However, when activity 2 is utilized at least two units of output are produced when using two units of each input. Total input usage is the same in both cases, however, more production arises out of the second production process than by adding the output of the first and third process. Thus, we get more out of using the inputs together,  $f(X + Y)$ , than we do separately,  $f(X) + f(Y)$ .

Does this violate the algorithmic assumptions? Within the model the production processes are strictly additive. Combination of any group of X's leads to an additive output effect. However, by utilizing different variables, a production process may be represented which is not strictly additive. Thus, one can usually handle nonadditive cases between variables by including "better" variables which are more productive (i.e.,  $X_2$  above). Nevertheless, the additivity assumption always holds for the individual

variables. It may not hold for the model through the combination of variables. This assumption is formally relaxed by the models covered in the nonlinear, price endogenous and risk chapters.

### 10.2.3 Uncertainty Examples

The certainty assumption may also be relaxed. Suppose we model a production process involving, a cost of \$3 in period 1 but that, in the second time period we are uncertain about how much of the product will be produced (e.g., harvested). Suppose that one of two uncertain events can occur in the second time period: no more than 2 units of the product may be sold for a price of \$5.00 with a probability of .3 or no more than 3 units of product could be sold at a price of \$4 with a probability of .7. This problem may be formulated as a classical so-called two-stage optimization problem (Dantzig, 1955). The formulation is

$$\begin{array}{rllll}
 \text{Max} & 3Y & + & .3(5X_1) & + & .7(4X_2) \\
 \text{s.t.} & Y & \leq & X_1 & & 0 \\
 & & & X_1 & \leq & 2 \\
 & Y & & & & X_2 & \leq & 0 \\
 & & & & & X_2 & \leq & 3 \\
 & Y, & X_1, & X_2 & \geq & 0.
 \end{array}$$

In this formulation a certain cost of \$3 is incurred when using variable Y. In turn, the production of Y permits sale under the two probabilistic events. The amounts sold are denoted  $X_1$  or  $X_2$  depending upon the event. Resources cannot be shifted between  $X_1$  and  $X_2$  (i.e., they are two mutually exclusive states of nature), thus, there are independent limits on the sale of  $X_1$  and  $X_2$ . However, Y precedes both. The objective function reflects the maximization of expected profits which are the expected revenue from sales less the cost of Y.

Thus, this formulation explicitly includes uncertainty. But, is the certainty assumption violated? Again, this formulation simultaneously satisfies and violates the algorithmic assumptions of LP. We have incorporated uncertainty within the formulation, but each variable contains certain coefficients. However, the overall model represents production under uncertainty. The uncertainty problem has been expressed in a problem where the model is certain of the uncertainty. Additional certainty assumption relaxations are discussed in the risk chapter. The specific example above is a sequential uncertainty, discrete stochastic or two-stage stochastic programming with recourse problem.

### 10.2.4 Noncontinuous Example

The continuity assumption when violated involves decision variables which are integer valued by nature (i.e., the number of cows, for instance). This may be relaxed by rounding when in the optimal solution

the integer variables have very large values. A problem of this type is as follows:

The solution without the requirement that  $X_1$  and  $X_2$  be integer is  $X_1 = 4,666 \frac{2}{3}$  and  $X_2 = 5,333 \frac{1}{3}$ . The model user might be willing to round this solution interpreting the solution as producing 4,667 of the first product and 5,333 of the second. This would clearly not be the optimal solution but might be practical and "close enough." Note, however, that the answer 4,667 and 5,333 slightly violates the second constraint. Nevertheless, decision makers might be willing to adopt this solution. In a practical problem this answer might even be interpreted as 4,700 and 5,300.

The continuity assumption is not practically relaxed other than by rounding large solution values or by solving an integer programming problem.

## References

- Baker, T. and B.A. McCarl, "Representing Farm Resource Availability over Time in Linear Programs: A Case Study." North Central Journal of Agricultural Economics, 4(1982):59-68.
- Dantzig, G.B. "Linear Programming Under Uncertainty." Management Science. 1(1955):197-206.
- Dantzig, G.D. and R.M. Van Slyke. "Generalized Linear Programming." Chapters in Optimization Methods for Large Scale Systems, D.A. Wisner, (ed.). McGraw-Hill: New York, 1970.
- Heady, E.O. and W.V. Candler. Linear Programming Methods. Ames, IA: Iowa State University Press, 1958.
- Lasdon, L. Optimization Theory for Large Systems. MacMillan: New York, 1970.
- McCarl, B.A. "Degeneracy, Duality, and Shadow Prices in Linear Programming." Canadian Journal of Agricultural Economics. 25(1977):70-73.
- McCarl, B.A., W.V. Candler, D.H. Doster, and P.R. Robbins. "Experiences with Farmer Oriented Linear Programming for Crop Planning." Canadian Journal of Agricultural Economics. 25(1977):17-30.
- Murtaugh, B. and M. Saunders. "MINOS 5.0 Users Guide." Technical Report SOL 83-20 Stanford University, 1983.
- Orchard-Hays, W. Advanced Linear Programming Computing Techniques. New York: McGraw-Hill, 1968.
- Schrage, L.E. Users Manual for LINDO. Palo Alto: The Scientific Press, 1981.



## CHAPTER XI: MULTI-OBJECTIVE PROGRAMMING

Optimization of a single objective oversimplifies the pertinent objective function in some potential mathematical programming application situations. Arguments can also be made following Simon that optimization is not as appropriate as satisficing. These two statements introduce the general topic of multiobjective programming. Multiobjective programming formally permits formulations where: a) solutions are generated which are as consistent as possible with target levels of goals; b) solutions are identified which represent maximum utility across multiple objectives; or c) solution sets are developed which contain all nondominated solutions. Multiple objectives can involve such considerations as leisure, decreasing marginal utility of income, risk avoidance, preferences for hired labor, and satisfaction of desirable, but not obligatory, constraints.

A discussion of this area requires some definitions. An objective is a measure that one is concerned about when making a choice among the decision variables (something to be maximized, minimized or satisfied like leisure, risk, profits, etc.). A goal implies that a particular goal target value has been chosen for an objective.

We will use "multiple objective programming" to refer to any mathematical program involving more than one objective regardless of whether there are goal target levels involved. Note, the literature contains conflicting definitions (see Blake and McCarl; Ignizio [1978,1983]; Romero [1989, 1991]). For example: a) goal programming has been used to refer to multiple objective problems with target levels; b) multiobjective programming has been used to refer to only the class of problems with weighted or unweighted multiple objectives; c) vector maximization has been used to refer to problems in which a vector of multiple objectives are to be optimized; and d) risk programming has been used to refer to multiobjective problems in which the objectives involve income and risk.

Multiobjective programming involves recognition that the decision maker is responding to multiple objectives. Generally, objectives are conflicting, so that not all objectives can simultaneously arrive at their optimal levels. An assumed utility function is used to choose appropriate solutions. Several fundamentally different utility function forms have been used in multiobjective models. These may be divided into three classes: lexicographic, multi-attribute utility and unknown utility.

The lexicographic utility function specification assumes the decision maker has a strictly ordered preemptive preference system among objectives with fixed target levels. For example, a lexicographic

system could have its first priority goal as income of not less than \$10,000; the second priority as leisure of not less than 20 hours a week; the third as income of no less than \$12,000, etc. This formulation is typical of "goal programming models." (Charnes and Cooper (1961); Lee). The various goals are dealt with in strict sequential order - higher goals before lower order goals. Once a goal has been dealt with (meeting or failing to meet the target level), its satisfaction remains fixed and the next lower order goal is considered. Consideration of the lower level goals does not alter the satisfaction of higher level goals and cannot damage the higher level goals with respect to target level attainment.

Multi-attribute utility approaches allow tradeoffs between objectives in the attainment of maximum utility. The most common form involves maximization of the sum of linearly weighted objectives. This type of formulation has been used by Candler and Boeljhe; and Barnett, Blake and McCarl.

The third utility approach involves an unknown utility function assumption. Here the entire Pareto efficient (nondominated) solution set is generated so that every solution is reported wherein one of the multiple objectives is as satisfied as it possibly can be without making some other objective worse off (Steur, Geoffrion (1968)).

### 11.1 Formulations

All of the above utility functions can be expressed in terms of the following problem. Assume there are multiple objectives which are given by

$$GX$$

where there are J decision variables (X) and R objectives. Thus, the matrix G is of dimension R by J while X is J by 1. These objectives can also be expressed in summation notation as

$$\sum_j g_{rj} X_j \quad \text{for all } r$$

When target levels are added, the objectives become

$$GX \leq T$$

The general goal programming problem, then, is as follows:

Select X so that we

$$\begin{array}{ll}
\text{optimize or achieve} & GX \\
\text{s.t.} & AX \# b \\
\text{Possibly subject to} & GX \$ T \\
& X \$ 0
\end{array}$$

Here the normal LP objective function is replaced by a more general function which permits use of different utility function forms (it is difficult to write the Pareto utility function in this form). The problem involves selection of the X's. The selection is driven by either optimization of some weighted tradeoff of objectives or through lexicographic achievement of various goal target levels. The specific formulations used for each of the above utility function specifications are given below.

### 11.1.1 Lexicographic Utility - Target Values

Perhaps the first application of multiobjective programming was the Charnes and Cooper goal programming formulation. The formulation is: Select X so that

$$AX \# b$$

$$X \$ 0$$

and so that the goals are handled in the following priority order:

$$\sum_j g_{1j} X_j \$ T_1$$

then

$$\sum_j g_{2j} X_j \$ T_2$$

on through to

$$\sum_j g_{Rj} X_j \$ T_R$$

for the R<sup>th</sup> and last goal.

The lexicographic multiple objective formulation is not precisely a LP problem. It has many structural characteristics in common with a LP problem; however, a conventional objective function is not defined, nor can a single LP formulation reflect imposition of the sequential ordering of the goals. Rather, an iterative procedure is needed (Lee). Essentially, the approach is to solve problems for each of the goals sequentially. When considering the i<sup>th</sup> goal solve the problem

$$\begin{array}{ll}
\text{Min} & w_i \\
\text{s.t.} & w_r \leq g_r - T_r \quad \text{for all } r \\
& g_r - g_{rj} X_j \leq 0 \quad \text{for all } r \\
& \sum_j a_{mj} X_j \leq b_m \quad \text{for all } m \\
& w_r \leq w_r^* \quad \text{for all } r < i \\
& w_r \leq 4 \quad \text{for all } r \leq i \\
& w_r, X_j \geq 0 \quad \text{for all } j \text{ and } r \\
& g_r \text{ unrestricted for all } r
\end{array}$$

The new variable  $w_r$  gives the amount that the goal level ( $g_{rj}X_j$ ) is less than the target value ( $T_r$ ), while  $g_r$  is the current level of goal  $r$ . When  $i = 1$  the problem minimizes the shortfall from the first goal target level, subject to the LP constraints. One of two solution situations will then occur. Either the optimum value of  $w_1$  (denoted  $w_1^*$ ) equals 0, indicating full satisfaction of the first goal, or  $w_1^* > 0$ , indicating the goal cannot be fully satisfied. Subsequently, a second problem is solved. This problem is virtually identical to the first, except  $w_2$  is minimized and a constraint is appended indicating that  $w_1$  cannot be any worse than the optimum value obtained at the end of the solution of the first stage ( $w_1^*$ ). This requires: 1) if goal 1 was met before, then goal 1 will continue to be met (i.e.,  $w_1$  must be less than or equal to zero); or 2) if goal 1 was not met, then the deviation from goal 1 will not get bigger than the minimum deviation obtained at the previous iteration. Thus, the prior objective is constrained to be no worse off than it was before. This problem, in effect, explores alternative optimums where we hold the prior objectives at their optimum values, then try to optimize the satisfaction of the subsequent objectives.

This procedure is executed for all  $R$  goals where different deviation variables are minimized at each stage and a constraint is added holding all previous deviations to maximum values prohibiting the earlier objectives from becoming worse off. Lee presents a more comprehensive discussion of the procedure while the example below gives an empirical application.

### 11.1.2 Utility Tradeoff Model

The second utility function type involves tradeoffs between various objectives. Such problems can be formulated as conventional linear programs. There have been two alternative formulations of this problem. They differ in their assumptions about target levels. The first formulation (appearing for example in Candler and Boeljhe) does not take into account target levels, maximizing the weighted sum of the

quantities of each objective. This is

$$\begin{aligned}
 & \text{Max} && \sum_r c_r q_r \\
 & \text{s.t.} && \sum_j g_{rj} X_j \geq gl_r && 0 && \text{for all } r \\
 & && gl_r \leq N_r q_r && 0 && \text{for all } r \\
 & && \sum_j a_{mj} X_j \leq b_m && && \text{for all } m \\
 & && X_j \geq 0, \quad q_r \geq 0 && && \text{for all } j \text{ and } r \\
 & && gl_r && \text{unrestricted} && \text{for all } r
 \end{aligned}$$

where  $c_r$  is the weight which expresses the importance of the  $r^{\text{th}}$  objective in the context of the decision maker's total utility and  $N_r$  is a normalizing factor which converts the goal values so they are valued somewhere around one.. The  $c_r$  coefficients would be in utility units per percent deviation from full satisfaction at the normalizing factor for the  $r^{\text{th}}$  objective achieved;  $gl_r$  is the amount of  $r^{\text{th}}$  objective in the optimal solution and  $q_r$  is the proportional satisfaction amount of  $r^{\text{th}}$  objective relative to the normalizing factor.

The objective function, maximizes multi-dimensional utility summed across all objectives. Each objective is weighted. The second equation sums the level of each objective into the variable  $gl_r$ . The third expresses satisfaction in terms of the normalizing factor. The fourth represents resource availability limitations, the fifth expresses nonnegativity constraints and the sixth allow the goal level to be positive or negative (note the normalizing factor must be of the appropriate sign).

The second weighted tradeoff formulation embodies goal target levels. The formulation is

$$\begin{aligned}
 & \text{Max} && \sum_r c_r^{\%} q_r^{\%} && \sum_r c_r^{\&} q_r^{\&} \\
 & \text{s.t.} && \sum_j g_{rj} X_j \geq gl_r && 0 && \text{for all } r \\
 & && gl_r \leq T_r q_r^{\%} && T_r q_r^{\&} && T_r && \text{for all } r \\
 & && \sum_j a_{mj} X_j \leq b_m && && && \text{for all } m \\
 & && X_j \geq 0, \quad q_r^{\%} \geq 0, \quad q_r^{\&} \geq 0 && && && \text{for all } j \text{ and } r \\
 & && gl_r && \text{unrestricted} && && \text{for all } r
 \end{aligned}$$

where  $T_r$  is the goal target value for objective  $r$ ,  $c_r^+$  is the weight attached to overachieving objective  $r$  relative to its target,  $c_r^-$  is the weight attached to underachieving objective  $r$  relative to its target,  $g_r$  is the level of the  $r$ th goal attained,  $q_r^+$  is the proportion that the target for objective  $r$  is overachieved, and  $q_r^-$  is the proportion that the target for objective  $r$  is underachieved.

This is again a linear program. The formulation is adapted from Lee and is used in Barnett, et al. (1982).

### 11.1.3 Unknown Utility Function

The other approach to multiobjective programming involves an unknown utility function assumption. Instead, the entire nondominated set of alternatives is generated. The formulation for this approach is exactly like the first one under the weighted tradeoff section above except that all possible weights are utilized in the problem. This particular approach has been studied extensively, (see, for example, the bibliographies in Steuer; and Ignizio, 1983) but does not appear to be very empirically useful.

## 11.2 Examples

A common example is used to demonstrate the above formulations. However, we will omit coverage of the unknown utility function model as its solutions would be rather extensive and its use has been limited.

The example builds upon the chair example used in the resource allocation section of the linear programming chapter. Suppose that the firm is interested in profit, idle labor, and idle lathe time in formulating its goals. Thus, the firm values leisure and slack lathe time as well as profits.

### 11.2.1 Lexicographic Formulation

The lexicographic formulation will be based on four different goals. The first goal is that the profit be greater than \$9,000, the second that idle labor be greater than or equal to 30 hours. The third is that idle lathe time be greater than or equal to 25 hours and the fourth, that profit be greater than or equal to \$9,500. The formulation of this problem with the deviation variables included is given in Table 11.1.

The approach in solving this problem is as above. First, the deviation from the \$9,000 profit target level will be minimized. This results in a deviation of zero. Subsequently, when minimizing the deviation from the idle labor goal, a constraint is entered allowing zero deviation from a profit level of \$9,000. Then we precede to consider idle lathe time holding the profits and idle labor goal achievement constant.

The GAMS implementation is given in Table 11.2 and file LEXICO. We have introduced profit accounting in line 67 through 69. Also, we have introduced slack variables accounting for idle resources

(lines 71-73). The four goals are modeled in lines 75 through 79 where the idle large and small lathe time is added and set equal to the goal level in line 79. In turn, line 80 relates the goal levels to their associated targets. Namely, the goal achievement level plus a deviation variable to make up the shortfall is set greater than or equal to the target. The constraints defined in line 82 then restrict the goal shortfall to be less than or equal to an acceptable deviation level.

The lexicographic approach is implemented in lines 85 through 96. The initial allowable deviations are set to a large number in line 30. The loop from lines 86 through 95 sets up the problem for each goal changing the objective function weights then solves. Subsequently, the maximum allowed deviation for future iterations is set equal to the shortfall. Report writing statements (lines 92-94) summarize the solution.

The solutions are presented in Table 11.3. Four solutions are involved, each arising when the goals are individually considered. In the first solution, the \$9,000 profit goal is easily attained while the labor and lathe time goals fall short with 4.024 units of idle labor and 17.073 units of idle lathe time. This plan has 12.195 units of functional normal chairs being produced and 108.337 units of fancy normal chairs.

In turn, when the idle labor problem is setup with the profit deviation restricted ( $w_1 \neq 0$ ), the solution exhibits profits of \$9,482 full attainment of the idle labor goal but the lathe goal is now 20.641 units short and we are \$18.421 short of the \$9,500 profit goal. This is achieved with production of 115.296 fancy chairs and 3.289 fancy chairs with maximum large lathe use.

Now turning to the idle lathe time problem, we constrain  $w_1 \neq 0$  and  $w_2 \neq 0$  then solve. Here we fully attain the \$9,000 profit and idle labor goals but fall 4.337 units short of the idle lathe time goal. Profit is now \$9,000. This plan is achieved by producing 15.152 units of functional normal chairs and 99.811 units of fancy normal chairs.

Finally, turning our attention to the last goal we find that we can not make any progress on it and have a solution which is equivalent to the solution in the step before.

The above results show the action of a lexicographic solution. Namely, the \$9,000 profit goal was satisfied and held satisfied throughout the process while the \$9,500 profit was not considered until the last step and the \$9,500 profit goal was only pursued when the other goals had been held at their satisfaction levels and as a consequence no progress could be made.

### **11.2.2 Weighted Tradeoff - No Targets**

One version of the weighted tradeoff formulation does not contain targets. We will follow the

theoretical formulation but will also include weights and normalizing factors. The normalizing factors are used so the goal magnitudes are approximately equal. Namely, we divide the profit goal through by 10,500, the labor goal by the labor resource availability and the total lathe goal by the lathe resource availability. This then will convert all of these goal numbers that range over 0 to 1 and allow relative weights to be used. In turn, the weight for profit equals one, while the weight for idle labor and lathe time both equal 0.4. The resultant formulation is in Table 11.4 and the GAMS instructions in Table 11.5 (file WEIGHT). Notice in the GAMS formulation the weight and normalization data are defined in lines 23-27, while the goal setup is essentially the same as in the previous example. The main variant is that the goal levels are normalized in lines 67-69 and the objective function is the sum of the goal weights times the goal levels (line 57).

The resultant solution to this problem is that 2.44 functional normal chairs are made while 112.2 fancy normal chairs are made and 6.829 fancy chairs with maximum use of the large lathe. The profit goal is achieved at a .92 level of the 10,500 normalization level (\$9,674) while the labor goal is achieved at a 0.217 level leaving 27.06 hours of labor idle. The idle lathe time variable equals 0.

### **11.2.3 Weighted Objective and Targets**

Finally, let us consider the unified target levels and weighted tradeoff formulation. Here we will use the same target levels as in the lexicographic model, but introduce weights where we value profit at a weight of one if it is more than the first goal and at -10 units if it is less than that. Idle labor and lathe time are weighted at 0.1 if they are greater than their target level and -0.4 if less. Finally, the profit in excess of \$9,500 is valued at .9 and less than that at -1. The goals are normalized by multiplying the deviation variables by the target value. This formulation is portrayed in Table 11.6 with the GAMS instructions in Table 11.7 (see file WTTAR). Notice in this formulation both positive and negative deviations are defined and the objective function both reflects shortfalls and excesses. The solution shows profit equals \$9,000, idle labor 25.08 units, and idle lathe time 25 units. This makes for a labor shortfall of .164 units and a profit shortfall of .05.

## **11.3 Choice Among Formulations**

An important question given the alternative formulations is: which one should be used for a problem? There are several general considerations involved in choosing among these formulations.

The first consideration is solver availability. Traditionally, the undominated approach requires a specially adapted solver. Such adaptations have been implemented (Steuer) although they are not routinely



available for more than small problems. In addition, the undominated set approach can be quite expensive computationally (Steuer). Thus, this criteria favors the weighted tradeoff or goal formulations where one could use standard algorithms.

Second, one needs to consider the required amount of decision maker contact, particularly if the model is being used for predictive purposes. The undominated solution alternative would not be satisfactory in a limited contact setting as it requires active choice of the "best" strategy by the decision maker. This would be particularly troubling in many predictive exercises as the methods would generate a large number of answers, any of which could be the solution depending on decision maker choice.

Third, the treatment of goal target levels is difficult in comparative static exercises. It is difficult to: 1) specify goal target levels and 2) conclude that the goal target levels do not depend upon the resource base. Certainly, an income goal is easier to satisfy if the resource base is augmented. Thus, the lexicographic utility function formulations are not scale neutral. Many LP models are built to do comparative static studies, such as what would happen if larger equipment were available, more land area, labor, etc. This causes difficulties in using comparative statics with the weighted tradeoff model using fixed goal target levels or the lexicographic model.

Fourth, weights are difficult to discover. A complex questioning or calculation procedure is required (we review procedures for establishing weights below). Further, the use of constant weights over the entire domain of goal levels may be questionable. Procedures for including diminishing marginal utility would involve quadratic or separable programs as covered in other chapters.

All things considered, we prefer the weighted tradeoff model due to its consistency with our perceptions that individuals are willing to establish tradeoffs between objectives on the margin and that most models will be used for comparative static analysis.

#### **11.4 Developing Utility Functions**

Conceptually, multiobjective programming problems look attractive. However, assuming one knows the objectives, it is difficult to specify the utility structure. Clearly, this is not a problem with the undominated solution procedure as there are no weights. On the other hand, one puts all the alternative solutions in the decision maker's lap, which could involve thousands of solutions. Here we address how to find the utility function for the other formulations.

The easiest system to use is the lexicographic system, where one has to establish goal targets and the preemptive order. Targets such as the minimum amount of debt service plus consumption or the desired

length of a vacation can be used. However, one must be careful in using these targets in comparative static analysis, as the relative ability to satisfy the targets changes with alterations in the resource base. Also, one must ask whether tradeoffs are in order.

Weights are more difficult and are the subject of the bulk of the discussion herein. The first way of specifying weights is to take decision makers' past actions and then through a grid search over alternative weights, choose weights so as to minimize deviations of the model solution from observed actions. An example is given by Brink and McCarl for a risk analysis problem. We know of no formal attempt to do this in other than a risk analysis framework. The advantage here is that one obtains weights which are somehow consistent with revealed preferences. However, in a LP problem there is a range of weights which will generate the same solution. It is therefore possible that the proper set of weights is somewhere within the range, but that the wrong set of weights is chosen. In turn, this set of weights could lead to dramatically different behavior in a comparative static study.

A third procedure involves survey techniques. Here decision makers are asked questions about the relative importance of objectives and then through a scaling procedure a set of objective weights is obtained. This was done by Barnett, Blake and McCarl; Smith and Capstick; and Harman, et al. Two difficulties arise with this procedure. First, there is no assurance that the surveys generate results which mimic actions in actual situations. Second, it is difficult to translate the results into the proper specification of the programming model objective.

The fourth procedure we discuss was proposed by Candler and Boehlje and applies to the weighted tradeoff models. The procedure involves interaction with the decision maker and is based on revealed preference. To begin the process, modelers choose an initial set of weights and present the answers to the decision maker. In turn, the decision maker expresses preference for a change in the objective satisfaction levels (i.e., the decision maker could argue that there was insufficient income and excessive risk). The modeler would then alter the weights on those objectives and rerun the model. The process would continue until the decision maker was satisfied. This approach has the advantage of obtaining goal weights consistent with the decision maker's preferences and the potential disadvantage of obtaining the proper output with the wrong set of goal weights, leading to improper solutions when one is doing further analysis.

Finally, we must comment that there is no real way to abstractly set up a multiple objective model. The weights for the multiple objectives clearly require interaction with the decision maker.

### **11.5 Shadow Prices**

Much discussion has been devoted in previous chapters to shadow prices but little here. In this section we explore the meaning of shadow prices in a weighted multiple objective problem and derive meaningful shadow prices.

The shadow prices for a weighted multiobjective problem nominally give the marginal change in the weighted utility of a marginal right hand side change. The weighted utility is a multi-dimensional utility measure constructed as the sum of the individual objectives times their weights. However, one must ask how useful it is to know the expected change in this multi-dimensional utility function. This ordinarily would probably not be terribly useful as decision makers will be more interested in knowing what happens to the specific objectives. Mathematically, the effect on the specific objectives may be derived as follows. The first weighted multiple objective function with the normalization factors set to one and the equality constraints substituted out becomes:

$$\begin{aligned} \text{Max } & \sum_j w_r (g_{rj} X_j) \\ \text{s.t. } & \sum_j a_{kj} X_j \leq b_k \text{ for all } k \\ & X_j \geq 0 \text{ for all } j \end{aligned}$$

Rearranging, we obtain

$$\begin{aligned} \text{Max } & \sum_j (\sum_r w_r g_{rj}) X_j \\ \text{s.t. } & \sum_j a_{kj} X_j \leq b_k \text{ for all } k \\ & X_j \geq 0 \text{ for all } j \end{aligned}$$

Note that each variable  $X_j$  has an objective function coefficient which is the sum of the weights times its relative goal contributions. This can be collapsed using

$$C_j = \sum_r (w_r g_{rj})$$

or, in matrix terms,

$$C = WG$$

We now turn our attention to shadow prices, given by

$$U = C_B B^{-1}$$

The  $C_B$  terms within the multiobjective programming model are given by the multiplication of goal weights times the goal levels involved with the basic variables

$$C_B^{-1} W G_B^{-1} [w_1 \ w_2 \dots \ w_r]$$

$$\begin{bmatrix} g_{11}^b & \dots & g_{1M}^b \\ g_{21}^b & \dots & g_{2M}^b \\ \dots & & \dots \\ g_{R1}^b & \dots & g_{RM}^b \end{bmatrix}$$

where the superscript b on the g terms refer to the coefficients associated with the basic variable in the various objectives.

The shadow price term can be rewritten as

$$U = C_B B^{-1} = W G_B B^{-1}$$

Here, the term  $G_B B^{-1}$  gives an unweighted set of shadow prices, each column of  $G_B B^{-1}$  shows how each objective function is affected by right hand side changes. Practically, these shadow prices could be obtained analytically from small problems and from larger problems by employing the solver starting from the optimal solution which can do a pricing pass but does not optimize (doing zero iterations).

### 11.5.1 Example

Suppose we were to maximize the following multiple goal objective problem and we are willing to assume that the weights are each 1. The problem with the composite objective

$$\text{Max} \begin{cases} 10X_1 \% X_2 \\ \& 7X_1 \% X_2 \end{cases}$$

$$\text{s.t.} \quad \begin{aligned} X_1 \& X_2 \# 5 \\ X_1 \% X_2 \# 10 \\ X_1 , X_2 \$ 0 \end{aligned}$$

function then becomes

$$\text{Max} \quad 1(10X_1 \% X_2) \% 1(\& 7X_1 \% X_2) \quad = \quad 3X_1 \% 2X_2$$

$$\text{s.t.} \quad \begin{aligned} X_1 \& X_2 \# 5 \\ X_1 \% X_2 \# 10 \\ X_1 , X_2 \$ 0 \end{aligned}$$

Solving this problem (see file SHADOW) we find that our solution consists  $X_1=7.5$  and  $X_2=2.5$ . equals 2.5. The basis matrix and its inverse are

$$B^{-1} = \begin{bmatrix} 1 & \& 1 \\ 1 & 1 \end{bmatrix} \quad B^{-1} = \begin{bmatrix} .5 & .5 \\ \& .5 & .5 \end{bmatrix}$$

The composite shadow prices are

$$C_B B^{-1} = \begin{bmatrix} 3 & 2 \end{bmatrix} \begin{bmatrix} .5 & .5 \\ .5 & .5 \end{bmatrix} = \begin{bmatrix} .5 & 2.5 \end{bmatrix}$$

However, if we break this down we get

$$W G_B B^{-1} = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 10 & 1 \\ 7 & 1 \end{bmatrix} \begin{bmatrix} .5 & .5 \\ .5 & .5 \end{bmatrix} = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 4.5 & 5.5 \\ 4 & 3 \end{bmatrix}$$

where the last matrix gives the shadow prices in terms of individual objectives. Thus, the change of one unit in the right hand side will increase the first objective by 4.5 while decreasing the second objective by 4. These shadow prices are more meaningful than the weighted shadow prices as they tell the implications of resource changes for each objective. Note that the weighted problem shadow prices are simply the individual weights times the shadow prices of this problem.

Preckel et al. proposed an alternative where they estimate relevant shadow prices by dividing through the by shadow prices on the individual objective accounting rows. However, this does not work as well as the theory above, as the shadow prices will be strictly proportional across the goals which need not happen in the general case (i.e. in the example above).

## References

- Barnett, D., B. Blake and B.A. McCarl. "Goal Programming via Multidimensional Scaling Applied to Sengalese Subsistence Farms." American Journal of Agricultural Economics. 64 (1982):720-27.
- Blake, B.F. and B.A. McCarl. "Goal Programming via Multidimensional Scaling Applied to Sengalese Subsistence Farming: A Reply." American Journal of Agricultural Economics. 65 (1983):832-33.
- Brink, L. and B.A. McCarl. "The Expected Income-Risk Trade-Off Among Corn Belt Crop Farmers." American Journal of Agricultural Economics. 60(1978):259-63.
- Candler, W. and M. Boehlje. "Use of Linear Programming in Capital Budgeting with Multiple Goals." American Journal of Agricultural Economics. 53(1977):325-330.
- Charnes, A. and W.W. Cooper. Management Models and Industrial Applications of Applications of Linear Programming. Vol. I. New York: John Wiley & Sons, 1961.
- Charnes, A., W.W. Cooper and R.O. Ferguson. "Optimal Estimation of Executive Compensations by Linear Programming." Management Science. 1(1955):138-51.
- Geoffrion, A.M. "Proper Efficiency and the Theory of Vector Maximization." Journal of Mathematical Analysis and Applications. 22(1968):618-30.
- Harman, W., R. Hatch, V. Eidman and P. Claypool. "An Evaluation of Factors Affecting the Hierarchy of Multiple Goals." Technical Bulletin T-134, Oklahoma State Agriculture Experiment Station and USDA, June 1972.
- Ignizio. "A Review of Goal Programming: A Tool for Multiobjective Analysis." Journal of Operational Research Society. 29(1978):1109-1119.
- \_\_\_\_\_. "Generalized Goal Programming: An Overview." Computers and Operations Research. 10(1983):277-89.
- Lee, S.M. Goal Programming for Decision Analysis. Philadelphia: Auerbach, 1972.
- Preckel, P.V., A.M. Featherstone, and T.G. Baker. "Interpreting Dual Variables for Optimization with Nonmonetary Objectives." American Journal of Agricultural Economics. 69(1988):849-851.
- Romero, C. Handbook of Critical Issues in Goal Programming. Oxford, New York: Pergamon Press, 1991.
- Romero, C. Multiple Criteria Analysis for Agricultural Decisions. Amsterdam, New York: Elsevier, 1989.
- Simon, H.A. Models of Man. New York: John Wiley and Sons, 1954.
- Smith, D. and D. Capstick. "Establishing Priorities Among Multiple Management Goals." Southern Journal of Agricultural Economics. 8(1972):37-43.
- Steuer, R. "Vector Maximization Gradient Core Contraction Techniques." in Multiple Criteria Problem Solving. S. Zionts, ed. Berlin: Springer-Verlag, 1978, pp. 462-81.

**Table 11.1. Tableau for Lexicographic Example**

Objective	Original Decision Variables						Idle Resources			Goal Levels			Goal Deviations			RHS		
	X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	X <sub>5</sub>	X <sub>6</sub>	Lrg Lathe	Sml Lathe	Carver Labor	Profit 9000	Idle Labor	Idle Lathe	Profit 9500	Prof 9000	Idle Labor		Idle Lathe	Profit 9500
Original Problem Equations																		
Profit	67	66	66.3	80	78.5	78.4												Min
Small Lathe	0.8	1.3	0.2	1.2	1.7	0.5	1											= 0
Large Lathe	0.5	0.2	1.3	0.7	0.3	1.5		1										= 140
Carver	0.4	0.4	0.4	1.0	1.0	1.0			1									= 90
Labor	1.0	1.05	1.1	0.8	0.82	0.84				1								= 120
Goal																		= 125
Profit 9000																		= 0
Idle Labor										1								= 0
Idle Lathe											1							= 0
Profit 9500																		= 0
Goal																		= 0
Profit 9000														1				\$ 9000
Idle Labor															1			\$ 30
Idle Lathe																1		\$ 25
Profit 9500																	1	\$ 950
Lexicographic Satisfaction																		# 0
Profit														1				# 999
Idle															1			# 999
Idle																1		# 999
Profit																	1	# 999

**Table 11.2. GAMS Formulation of Lexicographic Example**

```

5  SET          GOALS          GOALS IN THE ORDER THEY ARE TO BE MET
6                                     /PROFIT1,LABOR,LATHETIME,PROFIT2/
7          PROCESS          TYPES OF PRODUCTION PROCESSES
8                                     /FUNCTNORM , FUNCTMXSML , FUNCTMXLRG
9                                     ,FANCYNORM , FANCYMXSML , FANCYMXLRG/
10         RESOURCE          TYPES OF RESOURCES
11                                     /SMLLATHE,LRGLATHE,CARVER,LABOR/
13  ALIAS(GOALS,GOAL) ;
15  PARAMETER PRICE(PROCESS)      PRODUCT PRICES BY PROCESS
16                                     /FUNCTNORM 82, FUNCTMXSML 82, FUNCTMXLRG 82
17                                     ,FANCYNORM 105, FANCYMXSML 105, FANCYMXLRG 105/
18  PRODCOST(PROCESS)      COST BY PROCESS
19                                     /FUNCTNORM 15, FUNCTMXSML 16 , FUNCTMXLRG 15.7
20                                     ,FANCYNORM 25, FANCYMXSML 26.5, FANCYMXLRG 26.6/
21  RESORAVAIL(RESOURCE)      RESOURCE AVAILABILITY
22                                     /SMLLATHE 140, LRGLATHE 90,
23                                     CARVER 120, LABOR 125/
24  TARGET(GOALS)          GOAL TARGET LEVELS
25                                     /PROFIT1 9000,LABOR 30,LATHETIME 25
26                                     ,PROFIT2 9500/
27  DEV(GOALS)      MAXIMUM DEVIATION BY GOAL
28  WEIGHTS(GOALS)      WEIGHTS BY GOAL ;
30  DEV(GOALS)=999999;
31  WEIGHTS(GOALS)=0.00001;
33  TABLE RESORUSE(RESOURCE,PROCESS)      RESOURCE USAGE
35          FUNCTNORM  FUNCTMXSML  FUNCTMXLRG
36  SMLLATHE          0.80      1.30      0.20
37  LRGLATHE          0.50      0.20      1.30
38  CARVER            0.40      0.40      0.40
39  LABOR             1.00      1.05      1.10
40  +          FANCYNORM  FANCYMXSML  FANCYMXLRG
41  SMLLATHE          1.20      1.70      0.50
42  LRGLATHE          0.70      0.30      1.50
43  CARVER            1.00      1.00      1.00
44  LABOR             0.80      0.82      0.84;
46  POSITIVE VARIABLES
47          PRODUCTION(PROCESS)      ITEMS PRODUCED BY PROCESS
48          IDLE(RESOURCE)      SLACK VARIABLES FOR RESOURCES
49          GOALLEVEL(GOALS)      GOAL LEVELS
50          PROFIT          TOTALPROFIT
51          SHORTFALL(GOALS)      GOAL SHORTFALLS;
52  VARIABLES
53          GOALOBJ          GOAL OBJECTIVE;
54  EQUATIONS
55          OBJT          OBJECTIVE FUNCTION
56          PROFITACCT      PROFIT ACCOUNTING
57          AVAILABLE(RESOURCE)      RESOURCES AVAILABLE
58          IDLLABGOAL      IDLE LABOR GOAL
59          PROFITGL1      PROFIT1 GOAL
60          PROFITGL2      PROFIT2 GOAL
61          LATHEGOAL      IDLE LATHE GOAL
62          TARGS(GOALS)      GOAL TARGETS
63          MAXSHORT(GOALS)      SHORTFALL LIMITS;
64
65  OBJT.. GOALOBJ =E= SUM(GOALS,WEIGHTS(GOALS)*SHORTFALL(GOALS)) ;
66
67  PROFITACCT.. PROFIT =E=
68          SUM(PROCESS,(PRICE(PROCESS)-PRODCOST(PROCESS))
69          * PRODUCTION(PROCESS)) ;
70

```



```

71  AVAILABLE(RESOURCE)..
72      SUM(PROCESS,RESOURUSE(RESOURCE,PROCESS)*PRODUCTION(PROCESS))
73          +IDLE(RESOURCE) =E= RESORAVAIL(RESOURCE);
74
75  IDLLABGOAL..  IDLE("LABOR") =E= GOALLEVEL("LABOR") ;

```

**Table 11.2. GAMS Formulation of Lexicographic Example (Continued)**

---

```

76  PROFITGL1..  PROFIT      =E= GOALLEVEL("PROFIT1");
77  PROFITGL2..  PROFIT      =E= GOALLEVEL("PROFIT2");
78  LATHEGOAL..  IDLE("LRGLATHE")+IDLE("SMLLATHE")
79              =E= GOALLEVEL("LATHETIME");
80  TARGSGOALS.. GOALLEVEL(GOALS) + SHORTFALL(GOALS) =G= TARGET(GOALS) ;
81
82  MAXSHORT(GOALS).. SHORTFALL(GOALS) =L= DEV(GOALS);
83
84  MODEL RESALLOC /ALL/;
85  PARAMETER  GOALDATA(GOAL,*,*)
86  LOOP(GOAL,
87      WEIGHTS(GOAL)=1.
88
89      SOLVE RESALLOC USING LP MINIMIZING GOALOBJ;
90      DEV(GOAL)=SHORTFALL.L(GOAL);
91      WEIGHTS(GOAL)=0.00001;
92      GOALDATA(GOAL,GOALS,"ATTAIN")=GOALLEVEL.L(GOALS) ;
93      GOALDATA(GOAL,GOALS,"SHORT")=SHORTFALL.L(GOALS) ;
94      GOALDATA(GOAL,PROCESS,"XLEVEL")=PRODUCTION.L(PROCESS);
95      );
96  DISPLAY GOALDATA;

```

---

**Table 11.3. Solution to Lexicographic Example**

Goal Being Production Pursued	Solution Item	Idle	Goal Attainment	Goal Level	Goal Shortfall	Goal Level
PROFIT1	PROFIT1	GOAL	9500.000	9500	0	
	.LABOR	GOAL	25.976	30	4.024	
	.LATHETIME	GOAL	7.927	25	17.073	
	.PROFIT2	GOAL	9500.000	9500	0	
	.FUNCTNORM	PROD				
	.FANCYNORM	PROD				
12.195						
108.537						
LABOR	.PROFIT1	GOAL	9481.579	9500	0	
	.LABOR	GOAL	30.000	30	0	
	.LATHETIME	GOAL	4.359	25	20.641	
	.PROFIT2	GOAL	9481.579	9500	18.421	
	.FANCYNORM	PROD				
	.FANCYMXLRG	PROD				
115.296						
3.289						
LATHETIME	.PROFIT1	GOAL	9000.000	9500	0	
	.LABOR	GOAL	30.000	30	0	
	.LATHETIME	GOAL	20.663	25	4.337	
	.PROFIT2	GOAL	9000.000	9500	500.000	
	.FUNCTNORM	PROD				
	.FANCYNORM	PROD				
15.152						
99.811						
PROFIT2	.PROFIT1	GOAL	9000.000	9500	0	
	.LABOR	GOAL	30.000	30	0	
	.LATHETIME	GOAL	20.663	25	4.337	
	.PROFIT2	GOAL	9000.000	9500	500.000	
	.FUNCTNORM	PROD				
	.FANCYNORM	PROD				
15.152						
99.811						

**Table 11.4. Tableau for Weighted Tradeoff Example**

	Original Decision Variables						Profit	Idle Resources			Goal Levels			Goal Deviations			RHS
	X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	X <sub>5</sub>	X <sub>6</sub>		Lrg Lathe	Sml Lathe	Carver	Labor	Profit	Idle Labor	Idle Lathe	Profit	Idle Labor	
Objective																	Min
Original Problem Equations																	
Profit	67	66	66.3	80	78.5	78.4	-1										= 0
Small Lathe	0.8	1.3	0.2	1.2	1.7	0.5		1									= 140
Large Lathe	0.5	0.2	1.3	0.7	0.3	1.5			1								= 90
Carver	0.4	0.4	0.4	1.0	1.0	1.0				1							= 120
Labor	1.0	1.05	1.1	0.8	0.82	0.84					1						= 125
Goal																	
Profit							1										= 0
Idle Labor										1							= 0
Idle Lathe											1						= 0
Goal Level																	
Profit 9000															10500		= 0
Idle Labor																125	= 0
Idle Lathe																	230
																	= 0

**Table 11.5. GAMS Setup for Weighted Objective Example**

```

4
5 SET          GOALS          /PROFIT,LABOR,LATHETIME/
6              PROCESS        TYPES OF PRODUCTION PROCESSES
7                                  /FUNCTNORM , FUNCTMXSML , FUNCTMXLRG
8                                  ,FANCYNORM , FANCYMXSML , FANCYMXLRG/
9              RESOURCE        TYPES OF RESOURCES
10                                 /SMLLATHE,LRGLATHE,CARVER,LABOR/
11
12 PARAMETER PRICE(PROCESS)      PRODUCT PRICES BY PROCESS
13                               /FUNCTNORM 82, FUNCTMXSML 82, FUNCTMXLRG 82
14                               ,FANCYNORM 105, FANCYMXSML 105, FANCYMXLRG 105/
15          PRODCOST(PROCESS)    COST BY PROCESS
16                               /FUNCTNORM 15, FUNCTMXSML 16 , FUNCTMXLRG 15.7
17                               ,FANCYNORM 25, FANCYMXSML 26.5, FANCYMXLRG 26.6/
18          RESORAVAIL(RESOURCE) RESOURCE AVAILABILITY
19                               /SMLLATHE 140, LRGLATHE 90,
20                               CARVER 120, LABOR 125/
21          WEIGHTS(GOALS)        WEIGHT FOR GOALS
22                               /PROFIT 1,LABOR 0.4,LATHETIME 0.4/
23          MAGNITUDE(GOALS)      MAGNITUDE FOR GOALS
24                               /PROFIT 10500/;
25          MAGNITUDE("LATHETIME")=RESORAVAIL("SMLLATHE")+RESORAVAIL("LRGL
ATHE");
26          MAGNITUDE("LABOR")=RESORAVAIL("LABOR");
27
28
29 TABLE RESOURUSE(RESOURCE,PROCESS) RESOURCE USAGE
30
31              FUNCTNORM  FUNCTMXSML  FUNCTMXLRG
32 SMLLATHE      0.80      1.30      0.20
33 LRGLATHE      0.50      0.20      1.30
34 CARVER        0.40      0.40      0.40
35 LABOR         1.00      1.05      1.10
36 +            FANCYNORM  FANCYMXSML  FANCYMXLRG
37 SMLLATHE      1.20      1.70      0.50
38 LRGLATHE      0.70      0.30      1.50
39 CARVER        1.00      1.00      1.00
40 LABOR         0.80      0.82      0.84;
41
42 POSITIVE VARIABLES
43          PRODUCTION(PROCESS) ITEMS PRODUCED BY PROCESS
44          IDLE(RESOURCE)      SLACK VARIABLES FOR RESOURCES
45          GOALLEVEL(GOALS)    GOAL LEVELS
46          PROFIT              TOTALPROFIT;
47 VARIABLES
48          GOALOBJ              GOAL OBJECTIVE;
49 EQUATIONS
50          OBJT                  OBJECTIVE FUNCTION
51          PROFITACCT            PROFIT ACCOUNTING
52          AVAILABLE(RESOURCE)  RESOURCES AVAILABLE
53          IDLLABGOAL           IDLE LABOR GOAL
54          PROFITGOAL           PROFIT GOAL
55          LATHEGOAL            IDLE LATHE GOAL;
56
57 OBJT.. GOALOBJ =E= SUM(GOALS,WEIGHTS(GOALS)*GOALLEVEL(GOALS)) ;
58
59 PROFITACCT.. PROFIT =E=
60          SUM(PROCESS,(PRICE(PROCESS)-PRODCOST(PROCESS))
61              * PRODUCTION(PROCESS)) ;
62

```

```

63  AVAILABLE(RESOURCE)..
64      SUM(PROCESS,RESOURUSE(RESOURCE,PROCESS)*PRODUCTION(PROCESS))
65          +IDLE(RESOURCE) =E= RESORAVAIL(RESOURCE);
66
67  IDLLABGOAL..  IDLE("LABOR") =E= GOALLEVEL("LABOR")*MAGNITUDE("LABOR");
68  PROFITGOAL..  PROFIT          =E=
GOALLEVEL("PROFIT")*MAGNITUDE("PROFIT");
69  LATHEGOAL..  IDLE("LRGLATHE")+IDLE("SMLLATHE")
70              =E= GOALLEVEL("LATHETIME")*MAGNITUDE("LATHETIM
E");
71  MODEL RESALLOC /ALL/;
72  SOLVE RESALLOC USING LP MAXIMIZING GOALOBJ;

```

---

**Table 11.6. Tableau for Weighted Tradeoff with Targets Example**

Equations	Original Decision Variables						Idle Resources			Goal Levels			Goal Deviations						RHS								
	Profit						Small Lathe	Large Lathe	Carv Labor	Profit 9000	Idle Labor	Idle Lathe	Profit 9000	Idle Labor	Idle Lathe	Profit 9500	Idle Labor	Idle Lathe		Profit 9500							
	X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	X <sub>5</sub>	X <sub>6</sub>				+	-	+	+	-	-	+	+	-		-							
Objective																				Max							
Original Problem	67	66	66.3	80	78.5	78.4																					
	0.8	1.3	0.2	1.2	1.7	0.5	1															=	0				
	0.5	0.2	1.3	0.7	0.3	1.5		1															=	140			
	0.4	0.4	0.4	1.0	1.0	1.0			1															=	90		
	1.0	1.05	1.1	0.8	0.82	0.84				1														=	120		
																									=	125	
Goal Level Identity																									=	0	
																										=	0
																										=	0
																										=	0
Goal Satisfaction																										=	9000
																										=	30
																										=	25
																										=	9500

**Table 11.7. GAMS Setup for Weighted Objective with Target Example**

```

4
5 SET          GOALS          GOALS IN THE ORDER THEY ARE TO BE MET
6                               /PROFIT1,LABOR,LATHETIME,PROFIT2/
7           PROCESS          TYPES OF PRODUCTION PROCESSES
8                               /FUNCTNORM , FUNCTMXSML , FUNCTMXLRG
9                               ,FANCYNORM , FANCYMXSML , FANCYMXLRG/
10          RESOURCE        TYPES OF RESOURCES
11                               /SMLLATHE,LRGLATHE,CARVER,LABOR/
12          DIR              GOAL DEVIATION DIRECTION /MORETHAN,
LESSTHAN/
13
14  ALIAS(GOALS,GOAL) ;
15
16  PARAMETER PRICE(PROCESS)   PRODUCT PRICES BY PROCESS
17                               /FUNCTNORM 82, FUNCTMXSML 82, FUNCTMXLRG 82
18                               ,FANCYNORM 105, FANCYMXSML 105, FANCYMXLRG 105/
19          PRODCOST(PROCESS)  COST BY PROCESS
20                               /FUNCTNORM 15, FUNCTMXSML 16 , FUNCTMXLRG 15.7
21                               ,FANCYNORM 25, FANCYMXSML 26.5, FANCYMXLRG 26.6/
22  RESORAVAIL(RESOURCE) RESOURCE AVAILABILITY
23                               /SMLLATHE 140, LRGLATHE 90,
24                               CARVER 120, LABOR 125/
25          TARGET(GOALS)     GOAL TARGET LEVELS
26                               /PROFIT1 9000,LABOR 30,LATHETIME 25
27                               ,PROFIT2 9500/
28          MAGNITUDE(GOALS)  MAGNITUDE FOR GOALS;
29          MAGNITUDE(GOALS)=TARGET(GOALS);
30
31  TABLE          WEIGHTS(GOALS,dir)   WEIGHTS BY GOAL
32
33                               MORETHAN   LESSTHAN
34          PROFIT1          1           -10
35          LABOR            0.1         -0.4
36          LATHETIME        0.1         -0.4
37          PROFIT2          .9          -1. ;
38
39  TABLE RESOURUSE(RESOURCE,PROCESS) RESOURCE USAGE
40
41          FUNCTNORM  FUNCTMXSML  FUNCTMXLRG
42  SMLLATHE          0.80        1.30        0.20
43  LRGLATHE          0.50        0.20        1.30
44  CARVER            0.40        0.40        0.40
45  LABOR             1.00        1.05        1.10
46  +
47          FANCYNORM  FANCYMXSML  FANCYMXLRG
48  SMLLATHE          1.20        1.70        0.50
49  LRGLATHE          0.70        0.30        1.50
50  CARVER            1.00        1.00        1.00
51  LABOR             0.80        0.82        0.84;
52
53  POSITIVE VARIABLES
54  PRODUCTION(PROCESS) ITEMS PRODUCED BY PROCESS
55  IDLE(RESOURCE)      SLACK VARIABLES FOR RESOURCES
56  GOALLEVEL(GOALS)   GOAL LEVELS
57  PROFIT              TOTALPROFIT

```

57	SHORTFALL(GOALS)	GOAL SHORTFALLS
58	EXCESS(GOALS)	GOAL EXCESSES;
59	VARIABLES	
60	GOALOBJ	GOAL OBJECTIVE;
61	EQUATIONS	
62	OBJT	OBJECTIVE FUNCTION
63	PROFITACCT	PROFIT ACCOUNTING
64	AVAILABLE(RESOURCE)	RESOURCES AVAILABLE
65	IDLLABGOAL	IDLE LABOR GOAL
66	PROFITGL1	PROFIT1 GOAL
67	PROFITGL2	PROFIT2 GOAL

**Table 11.7. GAMS Setup for Weighted Objective with Target Example (Continued)**

---

```

68  LATHEGOAL      IDLE LATHE GOAL
69  TARGSGOALS)   GOAL TARGETS ;
70
71  OBJT.. GOALOBJ =E= SUM(GOALS,WEIGHTS(GOALS,"LESSTHAN")*SHORTFALL(GOALS)
72      +WEIGHTS(GOALS,"MORETHAN")*EXCESS(GOALS)) ;
73
74  PROFITACCT..  PROFIT =E=
75      SUM(PROCESS,(PRICE(PROCESS)-PRODCOST(PROCESS))
76      * PRODUCTION(PROCESS)) ;
77
78  AVAILABLE(RESOURCE)..
79      SUM(PROCESS,RESOURUSE(RESOURCE,PROCESS)*PRODUCTION(PROCESS))
80      +IDLE(RESOURCE) =E= RESORAVAIL(RESOURCE);
81
82  PROFITGL1..   PROFIT   =E= GOALLEVEL("PROFIT1");
83  IDLLABGOAL.. IDLE("LABOR") =E= GOALLEVEL("LABOR");
84  LATHEGOAL..  IDLE("LRGLATHE")+IDLE("SMLLATHE")
85      =E= GOALLEVEL("LATHETIME");
86  PROFITGL2..  PROFIT    =E= GOALLEVEL("PROFIT2");
87
88  TARGSGOALS)..
89      GOALLEVEL(GOALS) + MAGNITUDE(GOALS)*( SHORTFALL(GOALS) -EXCESS(GOALS))
90      =E= TARGET(GOALS) ;
91
92
93  MODEL RESALLOC /ALL/;
94
95  SOLVE RESALLOC USING LP MAXIMIZING GOALOBJ;

```

---



## CHAPTER XII: NONLINEAR OPTIMIZATION CONDITIONS

The previous material deals largely with linear optimization problems. We now turn our attention to continuous, certain, nonlinear optimization problems. The problems amenable to analysis using the methods in this chapter relax the LP additivity and proportionality assumptions.

The nonlinear optimization problem is important in a number of settings. This chapter will lay the ground work for several later chapters where price endogenous and risk problems are formulated as nonlinear optimization problems. Optimality conditions for the problems will be treated followed by brief discussion of solution principles.

### 12.1 Optimality Conditions

This section is devoted to the characterization of optimality conditions for nonlinear programming problems. These characterizations depend upon both first order conditions for identification of stationary points and second order conditions for discovery of the nature of the stationary points found. Consideration of types of optimum involves the topics of concavity and convexity. Thus, concavity and convexity are discussed. The presentation will not be extremely rigorous. Those interested in more rigorous treatments should consult books like Hadley, or Bazaraa and Shetty.

Nonlinear optimization problems may be constrained or unconstrained. Optimality conditions for unconstrained problems are ordinarily developed in calculus classes and will be briefly reviewed. Lagrangian multiplier and Kuhn Tucker based approaches are used to treat constrained problems and will be discussed here.

#### 12.1.1 Unconstrained Optimization

Unconstrained optimization is a topic in calculus classes. Such problems may contain one or N variables.

##### 12.1.1.1 Univariate

Problems with a single variable are called univariate. The univariate optimum for  $Y = f(X)$  occurs at points where the first derivative of  $f(X)$  with respect to  $X$  ( $f'(X)$ ) equals zero. However, points which have zero first derivatives do not necessarily constitute a minimum or maximum. The second derivative is used to discover character of a point. Points at which a relative minimum occurs

have a positive second derivative at that point while relative maximum occurs at points with a negative second derivative. Zero second derivatives are inconclusive.

It is important to distinguish between local and global optima. A local optimum arises when one finds a point whose value in the case of a maximum exceeds that of all surrounding points but may not exceed that of distant points. The second derivative indicates the shape of functions and is useful in indicating whether the optimum is local or global. The second derivative is the rate of change in the first derivative. If the second derivative is always negative (positive) that implies that any maximum (minimum) found is a global result. Consider a maximization problem with a negative second derivative for which  $f'(X^*)=0$ . This means the first derivative was  $> 0$  for  $X < X^*$  and was  $< 0$  for  $X > X^*$ . The function can never rise when moving away from  $X^*$  because of the sign of the second derivative. An everywhere positive second derivative indicates a global minimum will be found if  $f'(X^*)=0$ , while a negative indicates a global maximum.

#### 12.1.1.2 Multivariate functions

The univariate optimization results have multivariate analogues. In the multivariate case, partial derivatives are used, and a set of simultaneous conditions is established. The first and second derivatives are again key to the optimization process, excepting now that a vector of first derivatives and a matrix of second derivatives is involved.

There are several terms to review. First, the gradient vector,  $\nabla_x f(XE)$ , is the vector of first order partial derivatives of a multivariate function with respect to each of the variables evaluated at the point  $XE$ .

$$\nabla_x f(XE)_j = \left[ \frac{\partial f(XE)}{\partial X_j} \right]$$

where  $\partial f(XE) / \partial X_j$  stands for the partial derivative of  $f(X)$  with respect to  $X_j$  evaluated at  $XE$ , and  $XE$  depicts  $X_1E, X_2E, \dots, X_nE$ . The second derivatives constitute the Hessian matrix,

$$H(XE)_{ij} = \left[ \frac{\partial^2 f(XE)}{\partial X_i \partial X_j} \right]$$

The Hessian matrix, evaluated at  $XE$ , is an  $N \times N$  symmetric matrix of second derivatives of the function with respect to each variable pair.

The multivariate analogue of the first derivative test is that an  $XE$  must be found so that all terms

of the gradient vector simultaneously equal zero. The multivariate version of the second derivative test involves examination of the Hessian matrix at  $X^E$ . If that matrix is positive definite then the point  $X^E$  is a local minimum, whereas if the Hessian matrix is negative definite then the point is a local maximum. If the Hessian matrix is neither positive nor negative definite, then no conclusion can be made about whether this point is a maximum or minimum and one must conclude it is an inflection or saddle point.

### 12.1.2 Global Optima-Concavity and Convexity

The characterization of minimum and maximum points whether global or local is related to the concavity and convexity of functions. A univariate concave function has a negative second derivative everywhere and guarantees global maximum. A univariate convex function has a positive derivative everywhere yielding a global minimum. The multivariate analogues exhibit the proper definiteness of the Hessian matrix at all  $X$  points.

It is obviously desirable when dealing with optimization problems that global optimum be found. Thus, maximization problems are frequently assumed to be concave while minimization problems are assumed to be convex. Functions may also be locally concave or convex when the second derivative or Hessian only satisfies the sign convention in a region. Optimization problems over such functions can only yield local optimum.

Concavity of functions has been defined in another fashion. Concave functions exhibit the property that, given any two points  $X_1$  and  $X_2$  in the domain of the function, a line joining those points always lies below the function. Mathematically, this is expressed as

$$f(\delta X_1 + (1-\delta)X_2) \geq \delta f(X_1) + (1-\delta)f(X_2) \quad 0 \leq \delta \leq 1$$

Note that  $\delta f(X_1) + (1-\delta)f(X_2)$  is a line between  $f(X_1)$  and  $f(X_2)$  and that concavity requires this line to fall below the true function ( $f(\delta X_1 + (1-\delta)X_2)$ ) everywhere below this function.

Similarly, a line associated with two points on a convex function must lie above the true function

$$f(\delta X_1 + (1-\delta)X_2) \geq \delta f(X_1) + (1-\delta)f(X_2)$$

$$0 \leq \delta \leq 1$$

Concavity and convexity occur locally or globally. A function is globally concave if the conditions hold for all X or is locally concave or convex if the functions satisfy the conditions in some neighborhood.

The optimality conditions may be restated in terms of concavity and convexity. Namely, a multivariate function for which a stationary point  $X^*$  has been discovered has: a) a local maximum at  $X^*$  if the function is locally concave, b) a global maximum if the function is concave throughout the domain under consideration, c) a local minimum at  $X^*$  if the function is locally convex, d) a global minimum at  $X^*$  if the function is strictly convex, and e) a saddle point if the function is neither concave nor convex. At the stationary point, concavity and convexity for these conditions may be evaluated either using the two formulas above or using the positive or negative definiteness properties of the Hessian.

### 12.1.3 Constrained Optimization

The second major type of optimization problem is the constrained optimization problem. Two types of constrained problems will be considered: those subject to equality constraints without sign restricted variables and those subject to inequality constraints and/or sign restrictions on the variables. The optimality conditions for equality constrained optimization problems involve the Lagrangian and associated optimality conditions. The solution of problems with inequality constraints and/or variable sign restrictions relies on Kuhn-Tucker theory.

#### 12.1.3.1 Equality Constraints - The Lagrangian

Consider the problem

$$\begin{aligned} &\text{Maximize} && f(X) \\ &\text{s.t.} && g_i(X) = b_i \quad \text{for all } i \end{aligned}$$

where  $f(X)$  and  $g_i(X)$  are functions of  $N$  variables and there are  $M$  equality constraints on the problem.

Optimization conditions for this problem were developed in the eighteenth century by Lagrange. The Lagrangian approach involves first forming the function,

$$L(X, \delta) = f(X) + \sum_i \delta_i (g_i(X) - b_i)$$

where a new set of variables ( $\delta_i$ ) are entered. These variables are called Lagrange multipliers. In turn the problem is treated as if it were unconstrained with the gradient set to zero and the Hessian examined. The gradient is formed by differentiating the Lagrangian function  $L(X, \delta)$  with respect to both  $X$  and  $\delta$ . These resultant conditions are

$$\frac{\partial L}{\partial X_j} = \frac{\partial f(X)}{\partial X_j} - \sum_i \delta_i \frac{\partial g_i(X)}{\partial X_j} = 0 \quad \text{for all } j$$

$$\frac{\partial L}{\partial \delta_i} = (g_i(X) - b_i) = 0 \quad \text{for all } i.$$

In words, the first condition requires that at  $X^E$  the gradient vector of  $f(X)$  minus the sum of  $\delta_i$  times the gradient vector of each constraint must equal zero. The second condition says that at  $X^E$  the original constraints must be satisfied with strict equality. The first order condition yields a system of  $N+M$  equations which must be simultaneously satisfied. In this case, the derivatives of the objective function are not ordinarily driven to zero. Rather, the objective gradient vector is equated to the Lagrange multipliers times the gradients of the constraints.

$$\begin{aligned} \text{Max } & \sum_j c_j X_j \\ \text{s.t. } & \sum_j a_{ij} X_j = b_i \quad \text{for all } i \end{aligned}$$

These conditions are analogous to the optimality conditions of an LP consider a LP problem with  $N$  variables and  $M$  binding constraints. The first order conditions using the Lagrangian would be

$$\frac{\partial L}{\partial X_j} = c_j - \sum_i \delta_i a_{ij} = 0 \quad \text{for all } j$$

$$\frac{\partial L}{\partial \delta_i} = (\sum_j a_{ij} X_j - b_i) = 0 \quad \text{for all } i$$

Clearly, this set of conditions is analogous to the optimality conditions on the LP problem when one eliminates the possibility of zero variables and nonbinding constraints. Further, the Lagrange multipliers are analogous to dual variables or shadow prices, as we will show below.

Use of the Lagrangian is probably again best illustrated by example. Given the problem

$$\begin{aligned} \text{Minimize } & X_1^2 + X_2^2 \\ \text{s.t. } & X_1 + X_2 = 10 \end{aligned}$$

the Lagrangian function is

$$L(X, \lambda) = X_1^2 + X_2^2 - \lambda(X_1 + X_2 - 10)$$

Forming the Lagrange multiplier conditions leads to

$$\begin{aligned} \frac{\partial L}{\partial X_1} &= 2X_1 - \lambda = 0 \\ \frac{\partial L}{\partial X_2} &= 2X_2 - \lambda = 0 \end{aligned}$$

In turn, utilizing the first two conditions we can solve for  $X_1$  and  $X_2$  in terms of  $\lambda$  and getting

$$X_1 = X_2 = \lambda/2$$

Then plugging this into the third equation leads to the conclusion that

$$\lambda = 10; \quad X_1 = X_2 = 5$$

This is then a stationary point for this problem and, in this case, is a relative minimum. We will discuss the second order conditions below.

#### 12.1.3.1.1 Second Order Conditions - Constraint Qualifications: Convexity and Concavity

The Lagrangian conditions develop conditions for a stationary point, but yield no insights as to its nature. One then needs to investigate whether the stationary point is, in fact, a maximum or a minimum. In addition, the functions must be continuous with the derivatives defined and there are constraint qualifications which insure that the constraints are satisfactorily behaved.

Distinguishing whether a global or local optimum has been found, again, involves use of second order conditions. In this case, second order conditions arise through a "bordered" Hessian. The bordered Hessian is

$$H(X, \lambda) = \begin{bmatrix} 0 & \frac{\partial g_i(X)}{\partial X_j} \\ \frac{\partial g_i(X)}{\partial X_j} & \frac{\partial^2 f(X)}{\partial X_i \partial X_j} \end{bmatrix}$$

For original variables and  $m < n$  constraints, the stationary point is a minimum if starting with the principal minor of order  $2m + 1$  the last  $n - m$  principal minor determinants follow the sign  $(-1)^m$ . As similarly, if those principal minor determinants alternate in sign, starting with  $(-1)^{m+1}$ , then the stationary

point is a maximum. Mann originally developed this condition while Silberberg and Taha (1992) elaborate on it.

For the example above the bordered Hessian is

$$H(X^E, \theta^E) = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 2 & 0 \\ 1 & 0 & 2 \end{bmatrix}$$

Here, there are two variables and one constraint thus  $n - m = 2 - 1 = 1$ , and we need to examine only one determinant. This determinant is positive, thus  $X^E$  is a minimum.

An additional set of qualifications on the problem have also arisen in the mathematical programming literature. Here, the qualification involves relationship of the constraints to the objective function. It is expressed using the Jacobian matrix (J) which is defined with the elements

$$J = \begin{bmatrix} \frac{\partial g_i(X^*)}{\partial X_j} \end{bmatrix}_{ij}$$

This Jacobian matrix gives row vectors of the partial derivatives of each of the constraints with respect to the X variables. The condition for existence of  $\lambda$  is that the rank of this Jacobian matrix, evaluated at the optimum point, must equal the rank of the Jacobian matrix which has been augmented with a row giving the gradient vector of the objective function. This condition insures that the objective function can be written as a linear combination of the gradients of the constraints. Note that this condition does not imply that the Jacobian of the constraints has to be of full row rank. However, when the Jacobian of the constraints is not of full row rank, this introduces an indeterminacy in the Lagrange multipliers and is equivalent to the degenerate case in LP. Both Hadley and Pfaffenberger and Walker treat such cases in more detail.

The sufficient conditions for the Lagrangian also can be guaranteed by specifying: a) that the above rank condition holds which insures that the constraints bind the objective function, b) that the objective function is concave, and c) that the constraint set is convex. A convex constraint set occurs

when given any two feasible points all points in between are feasible.

### 12.1.3.1.2 Interpretation of Lagrange Multipliers

Hadley (1964) presents a useful derivation of the interpretation of Lagrange multipliers. We will follow this below. Assume  $Z$  is the optimal objective value, and  $X^*$  the optimal solution for the decision variables. Suppose now we wish to derive an expression for the rate at which the optimal objective function value changes when we change the right hand side. Then, by the chain rule, we obtain

$$\frac{\partial Z}{\partial b_i} = \sum_j \frac{\partial f(X^*)}{\partial X_j} \frac{\partial X_j}{\partial b_i}$$

If we also choose to differentiate the constraints with respect to  $b_i$ , we get

$$\frac{\partial g_k}{\partial b_i} = \sum_j \delta_{ik}^* \frac{\partial g_k(X^*)}{\partial X_j} \frac{\partial X_j}{\partial b_i}$$

where  $\delta_{ik}^*$  is the so-called Kronecker delta and equals

$$\delta_{ik}^* = \begin{cases} 1 & \text{if } i = k \\ 0 & \text{otherwise} \end{cases}$$

If we now take the equality relationship between the Kronecker  $\delta_{ik}^*$  and the derivatives of the constraints with respect to  $X_j^*$  we may rewrite this as

$$\delta_{ik}^* = \sum_{j=1}^N \frac{\partial g_k(X^*)}{\partial X_j} \frac{\partial X_j}{\partial b_i} = 0 \text{ for all } i \text{ and } k$$

Multiplying this through by  $\delta_k$ , we get

$$\delta_k \delta_{ik}^* = \sum_{j=1}^N \delta_k \frac{\partial g_k(X^*)}{\partial X_j} \frac{\partial X_j}{\partial b_i} = 0 \text{ for all } i \text{ and } k$$

Since the above term equals zero, we may freely sum over it and still obtain zero. If we add this term to the expression above we obtain

$$\frac{\partial Z}{\partial b_i} = \sum_{j=1}^N \frac{\partial f(X^*)}{\partial X_j} \frac{\partial X_j}{\partial b_i} + \sum_{k=1}^m \left( \delta_k \delta_{ik}^* + \sum_{j=1}^N \delta_k \frac{\partial g_k(X^*)}{\partial X_j} \frac{\partial X_j}{\partial b_i} \right) = 0 \text{ for all } i$$

Grouping terms yields



$$\frac{\partial Z}{\partial b_i} - \sum_k \delta_k \left( \frac{\partial f(X^*)}{\partial X_j} - \sum_k \delta_k \frac{\partial g_k(X^*)}{\partial X_j} \right) = 0 \quad \text{for all } i$$

The parenthetic part of this expression is equal to zero via the Lagrangian conditions. Thus, the sum over j always equals zero. The left hand side  $\delta_k$  times  $\frac{\partial f(X^*)}{\partial X_j}$  will be zero for all terms except where j equals k. Consequently, the sum equals  $\delta_i$ , and we obtain the conclusion

$$\frac{\partial Z}{\partial b_i} = \delta_i$$

or that the partial derivative of the objective function at optimality with respect to the  $i^{\text{th}}$  right hand side is equal to  $\delta_i$ . Thus, the  $\delta_i$ 's are analogous to shadow prices from ordinary LP. However, these are derivatives and are not generally constant over ranges of right hand side values as is true in LP. Rather, they are instantaneous projections of how the objective function would change given an infinitesimal change in the right hand side.

#### 12.1.3.2 Inequality Constraints - Kuhn Tucker Theory

Kuhn and Tucker, in 1951, developed optimality conditions for problems which contain inequality constraints and/or sign restricted variables. These conditions deal with the problem

$$\begin{array}{ll} \text{Maximize} & f(X) \\ \text{s.t.} & g(X) \leq b \\ & X \geq 0 \end{array}$$

The Kuhn-Tucker conditions state that if the following six conditions are satisfied then the solution

- 1)  $\nabla_x f(X^*) \quad \& \quad \delta^* \nabla_x g(X^*) \quad \# \quad 0$
- 2)  $[\nabla_x f(X^*) \quad \& \quad \delta^* \nabla_x g(X^*)] X^* \quad ' \quad 0$
- 3)  $X^* \quad \# \quad 0$
- 4)  $g(X^*) \quad \# \quad b$
- 5)  $\delta^* (g(X^*) \& b) \quad ' \quad 0$
- 6)  $\delta^* \quad \# \quad 0$

$X^*, \delta^*$  would be a candidate for optimality.

The conditions may be interpreted economically. The first condition requires that the first derivative of the objective function minus  $\delta^*$  times the first derivative of the constraints be less than or equal to zero at optimality. If one interprets the objective function as profit and the Lagrange multipliers as the cost of resources, and the constraint derivatives as the marginal resource requirements, then this condition requires that the marginal profit contribution of any product be less than or equal to the marginal cost of producing this product.

The second condition requires that the difference between the marginal profit and marginal cost times the X variable equals zero. The third condition requires nonnegative production. The second condition, taken together with the first and the third, requires that either the good be produced at a nonzero level and that marginal profit equals marginal cost, or the good not be produced and marginal profit be less than or equal to marginal cost (strictly less than in nondegenerate cases).

The fourth condition, in turn, requires that the original problem constraints be satisfied. The fifth condition requires that the Lagrange multiplier variables times the slack in the constraints equals zero, and the sixth condition that the Lagrange multipliers be nonnegative. The fourth and sixth conditions taken together, in conjunction with the fifth condition, require that either the constraint be binding and the Lagrange multiplier be nonzero (zero in degenerate cases), or that the Lagrange multiplier be zero and the constraint be nonbinding. Conditions 2 and 5 are analogous to the complementary slackness conditions in LP.

These conditions guarantee a global optimum, if the objective function is concave, the constraints  $g(X) \# b$  form a convex set, and one of several constraint qualifications occur. The simplest of these constraint qualifications require that the constraints form a convex set and a feasible interior point can

be found (the Slater condition). Another constraint qualification requires that the rank condition of the Jacobian be satisfied. There are other forms of constraint qualifications as reviewed in Bazaraa and Shetty; Gould and Tolle; and Peterson.

#### 12.1.3.2.1 Example 1

Consider the LP problem

$$\begin{array}{ll} \text{Maximize} & CX \\ \text{s.t.} & AX \leq b \\ & X \geq 0 \end{array}$$

The Kuhn-Tucker conditions of this problem are

- 1)  $C - \lambda' A \leq 0$
- 2)  $(C - \lambda' A)X = 0$
- 3)  $X \geq 0$
- 4)  $AX \leq b$
- 5)  $\lambda'(AX - b) = 0$
- 6)  $\lambda \geq 0$

These Kuhn-Tucker conditions are equivalent to the optimality conditions of the LP problem and show that the Kuhn-Tucker theory is simply a superset of LP theory and LP duality theory as the  $\lambda$ 's in the Kuhn-Tucker problem are equivalent to the LP dual variables.

#### 12.1.3.2.2 Example 2

The Kuhn-Tucker theory has also been applied to quadratic programming problems. A quadratic problem is

$$\begin{array}{ll} \text{Maximize} & CX + 1/2X'QX \\ \text{s.t.} & AX \leq b \\ & X \geq 0 \end{array}$$

and its Kuhn-Tucker conditions are

- 1)  $C^T X^* Q + \lambda^T (A X^* - b) = 0$
- 2)  $(C^T X^* Q + \lambda^T (A X^* - b))' = 0$
- 3)  $X^* \geq 0$
- 4)  $A X^* = b$
- 5)  $\lambda^T (A X^* - b) = 0$
- 6)  $\lambda \geq 0$

These Kuhn-Tucker conditions are close to a linear system of equations. If one disregards equations (2) and (5) the system is linear. These Kuhn-Tucker conditions have provided the basic equations that specialized quadratic programming algorithms (e.g. Wolfe) attempt to solve.

#### 12.1.4 Usage of Optimality Conditions

Optimality conditions have been used in mathematical programming for three purposes. The first and least used purpose is to solve numerical problems. Not many modelers check second derivatives or attempt to solve such things as the Kuhn-Tucker conditions directly. Rather, the more common usages of the optimality conditions are to characterize optimal solutions analytically, as is very commonly done in economics, or to provide the conditions that an algorithm attempts to achieve as in the Wolfe algorithm in quadratic programming.

### 12.2 Notes on Solution of Nonlinear Programming Models

Three general approaches have been used to solve nonlinear models. Problems have been approximated by a linear model and the resulting model solved via the simplex method as in the approximations chapter. Second, special problem structures (most notably those with a quadratic objective function and linear constraints) have been solved with customized algorithms. Third, general nonlinear programming algorithms such as MINOS within GAMS have been used.

A popular way of solving nonlinear programming problems is the "gradient" method (Lasdon and Waren, Waren and Lasdon). One of the popular gradient algorithms was developed by Murtaugh and Saunders (1987) and implemented in MINOS (which is the common GAMS nonlinear solver).

That method solves the problem

$$\begin{aligned} \text{Min } F(X) &= f(X^N) + CX^L \\ \text{s.t. } AX &\# b \\ X &\$ 0. \end{aligned}$$

where  $f(X)$  is a twice-differentiable convex function. Their approach involves an  $X$  vector which contains variables which only have linear terms,  $X^L$ , and variables with nonlinear objective terms,  $X^N$ .

MINOS first finds a feasible solution to the problem. The usual method employed in LP is to designate basic variables and non-basic variables which are set equal to zero. However, the optimal solution to a nonlinear problem is rarely basic. But Murtaugh and Saunders (1987) note that if the number of nonlinear variables is small, the optimal solution will be "nearly basic"; i.e., the optimal solution will lie near a basic solution. Thus, they maintain the traditional basic variables as well as superbasic and traditional non-basic variables. The superbasic variables have nonzero values with their levels determined by the first order conditions on those variables.

Given a current solution to the problem,  $X^0$ , MINOS seeks to improve the objective function value. The algorithm uses the gradient to determine the direction of change, thus GAMS automatically takes derivatives and passes them to MINOS. The algorithm proceeds until the reduced gradient of the objective function, in the space determined by the active constraints, is zero. MINOS can also solve problems with nonlinear constraints. See Gill, Murry and Wright for discussion.

### 12.3 Expressing Nonlinear Programs in Conjunction with GAMS

The solution of nonlinear programming problems in GAMS is a simple extension of the solution of linear programming problems in GAMS. One ordinarily has to do two things. First, one specifies the model using nonlinear expressions, and second the solve statement is altered so a nonlinear solver is used. In addition, it is desirable to specify an initial starting point and that the problem be well scaled.

An example quadratic programming problem is as follows:

$$\begin{array}{rcl}
 \text{Max} & 6Q_d & + 0.15Q_d^2 & + Q_s & + 0.1Q_s^2 \\
 & Q_d & & + Q_s & \# 0 \\
 & Q_d & & + Q_s & \# 0
 \end{array}$$

This problem is explained in the Price Endogenous modeling chapter and is only presented here for illustrative purposes. The GAMS formulation is listed in Table 12.1 and is called TABLE12.1 on the associated disk. The solution to this model as presented in Table 12.2 reveals shadow prices as well as optimal variable values and reduced costs. The SOLVE statement (line 34) uses the phrase "USING NLP" which signifies using nonlinear programming. Obviously users must have a license to a nonlinear programming algorithm such as MINOS to do this. Also, the objective function is specified as a nonlinear model in lines 26-28.

Finally a caution is also in order. Modelers should avoid nonlinear terms in equations to the extent possible (excepting in the equation expressing a nonlinear objective function). It is much more difficult for nonlinear solvers, like MINOS, to deal with nonlinear constraints.

## References

- Bazaraa, M.S. and C.M. Shetty. Nonlinear Programming: Theory and Algorithms. John Wiley and Sons, 1979.
- Gill, P.E., W. Murray, and M.H. Wright. Practical Optimization. Orlando, FL: Academic Press, Inc., 1981.
- Gould, F.J. and J.W. Tolle. "A Necessary and Sufficient Qualification for Constrained Optimization." SIAM Journal of Applied Mathematics 20(1971):164-172.
- Hadley, G. Nonlinear and Dynamic Programming. Reading, MA: Addison-Wesley Publishing Company, Inc., 1964.
- Kuhn, H. and A. Tucker. Nonlinear Programming, Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, ed. J. Neyman. Berkeley, CA: University of California Press, 1951, pp. 481-492.
- Lasdon, L.S. and A.D. Waren. "Survey of Nonlinear Programming Applications." Operations Research. 28(1980):1029-1073.
- Mann, H.B. "Quadratic Forms with Linear Constraints." Math Monthly. 50,7(1943):430-433.
- Murtagh, B.A. and M.A. Saunders. "Minos 5.1 User's Guide." Systems Optimization Lab; Department of Operations Research, Stanford University Technical Report SOL 83-20R. Revised January 1987.
- Peterson, D.W. "A Review of Constraint Qualifications in Finite-Dimensional Spaces." SIAM Review 15(1973):639-654.
- Pfaffenberger, R.C. and D.A. Walker. Mathematical Programming for Economics and Business. Ames, Iowa: Iowa State University Press, 1976.
- Silberberg, E. The Structure of Economics: A Mathematical Analysis. McGraw-Hill Book Company, 1978.
- Taha, H.A. Operations Research: An Introduction. New York, NY: Macmillan Publishing Company, 1992.
- Waren, A.D. and L.S. Lasdon. "The Status of Nonlinear Programming Software." Operations Research. 27(1979):431-456.
- Wolfe, P. "The Simplex Method for Quadratic Programming." Econometrica 27 (1959):382-398.

**Table 12.1. GAMS Formulation of Nonlinear Programming Example**

```

2
4 OPTION LIMCOL = 0;
5 OPTION LIMROW = 0;
6
7 SETS    CURVEPARM CURVE PARAMETERS /INTERCEPT,SLOPE/
8         CURVES    TYPES OF CURVES  /DEMAND,SUPPLY/
9
10 TABLE DATA(CURVES,CURVEPARM) SUPPLY DEMAND DATA
11
12         INTERCEPT  SLOPE
13 DEMAND      6      -0.30
14 SUPPLY      1      0.20
15
16 PARAMETERS SIGN(CURVES) SIGN ON CURVES IN OBJECTIVE FUNCTION
17           /SUPPLY -1, DEMAND 1/
18
19 POSITIVE VARIABLES QUANTITY(CURVES) ACTIVITY LEVEL
20
21 VARIABLES      OBJ          NUMBER TO BE MAXIMIZED
22
23 EQUATIONS      OBJJ          OBJECTIVE FUNCTION
24              BALANCE        COMMODITY BALANCE;
25
26 OBJJ.. OBJ =E= SUM(CURVES, SIGN(CURVES)*
27                  (DATA(CURVES,"INTERCEPT")*QUANTITY(CURVES)
28                  +0.5*DATA(CURVES,"SLOPE")*QUANTITY(CURVES)**2));
29
30 BALANCE.. SUM(CURVES, SIGN(CURVES)*QUANTITY(CURVES)) =L= 0 ;
31
32 MODEL PRICEEND /ALL/ ;
33
34 SOLVE PRICEEND USING NLP MAXIMIZING OBJ ;
35

```

**Table 12.2. Solution to Nonlinear Example Model**

Variables	Value	Reduced Cost	Equation	Level	Shadow Price
$Q_d$	10	0	Objective function	25	-
$Q_s$	10	0	Constraint	0	3



### CHAPTER XIII: PRICE ENDOGENOUS MODELING

A common economic application of nonlinear programming involves price endogenous models. In the standard LP model, input and output prices or quantities are assumed fixed and exogenous. Price endogenous models are used in situations where this assumption is felt to be untenable. Such problems can involve modeling an industry or sector such that the level of output or purchases of inputs is expected to influence equilibrium prices.

$$P_d = a_d - b_d Q_d.$$

The approach to formulating such problems was motivated by Samuelson, who suggested solving optimization problems whose first-order conditions constituted a system of equations characterizing an equilibrium. Suppose we follow this approach by first defining a system of equations, then posing the related optimization problem. Let an inverse demand equation be defined where  $P_d$  is price of the product,  $a_d$  is the intercept,  $b_d$  is the slope, and  $Q_d$  is the quantity demanded. Similarly, suppose we have an inverse supply equation

$$P_s = a_s + b_s Q_s,$$

where the terms are defined analogously. An equilibrium solution would have price and quantity equated and would occur at the simultaneous solution of the equations

$$P_d = P_s$$

or

$$a_d - b_d Q_d = a_s + b_s Q_s$$

and

$$Q_d = Q_s$$

One should also recognize some possible peculiarities of the equilibrium, namely it is possible that the markets could clear at zero quantity, in which case the supply price might be greater than or equal to the demand price.

Thus, we can write the condition that the equilibrium price ( $P^*$ ) is greater than or equal to the demand price

$$a_d - b_d Q_d \geq P^*$$

Simultaneously, the market price may be less than the supply price,

$$a_s - b_s Q_s \leq P^*$$

One can also argue that these two relations should only be inequalities when the quantity supplied or demanded equals zero. Namely, when the price of demand is less than the equilibrium price, then zero quantity should be demanded. Similarly, when the price of supply is greater than the equilibrium price, then zero quantity should be supplied. Simultaneously, when a non-zero quantity is supplied or demanded, then the equilibrium price should equal the supply or demand price. This relationship can be expressed through complementary slackness like relations where

$$\begin{aligned} (a_d - b_d Q_d - P^*) Q_d &= 0 \\ (a_s - b_s Q_s - P^*) Q_s &= 0 \end{aligned}$$

One should also recognize that the quantity supplied must be greater than or equal to the quantity demanded

$$Q_s \geq Q_d$$

but, if the quantity supplied is strictly greater than the quantity demanded, then the equilibrium price should be zero. Mathematically this relationship is

$$(Q_s - Q_d) P^* = 0$$

Finally, we state nonnegativity conditions for price and quantities,

$$Q_d, Q_s, P^* \geq 0.$$

The above equations are similar to the Kuhn-Tucker conditions. In particular, if  $P^*$  is taken to be a dual variable, then the above equation system is equivalent to the Kuhn-Tucker conditions of the following optimization model

$$\begin{aligned} \text{Max } & a_d Q_d - 1/2 b_d Q_d^2 - a_s Q_s + 1/2 b_s Q_s^2 \\ \text{s.t. } & Q_d - Q_s \leq 0 \\ & Q_d, Q_s \geq 0 \end{aligned}$$

where  $P^*$  is the dual variable associated with the first constraint. Optimizing this model solves our

equilibrium problem.

This is a quadratic programming problem. The formulation was originally motivated by Enke; and Samuelson. Later it was fully developed by Takayama and Judge (1973). The general form maximizes the integral of the area underneath the demand curve minus the integral underneath the supply curve (Figure 13.1), subject to a supply-demand balance. The resultant objective function value is commonly called consumers' plus producers' surplus.

The graphical representation allows one to develop a practical interpretation of the shadow price. Consider what happens if the  $Q_d - Q_s = 0$  constraint is altered so that the right hand side is one ( $Q_d - Q_s = 1$ ). In this case demand is allowed to be one unit greater than supply. Assuming the one unit is small relative to total quantity then we get an area increment that is approximately the height of the equilibrium price and one unit wide (Figure 13.2). The resultant objective function then is the original value plus an area equaling the equilibrium price. Thus, the change in the objective function when increasing the right hand side (the shadow price) can be interpreted as the equilibrium price. This also equals the Lagrange multiplier introduced when applying Kuhn-Tucker theory.

Example

Suppose we have

$$P_d = 6 - .3Q_d$$

$$P_s = 1 + .2Q_s$$

Then the formulation is

$$\begin{aligned} \text{Max } & 6Q_d - 0.15Q_d^2 - Q_s + 0.1Q_s^2 \\ & Q_d - Q_s = 0 \\ & Q_d, Q_s \geq 0 \end{aligned}$$

The GAMS formulation of this model is in Table 13.1 and file PRICEND. Note that there are two important changes in this setup compared to an LP. The first is that the objective function equation contains the nonlinear squared terms. The second is that in the SOLVE statement we indicate that the problem is a nonlinear programming problem by saying SOLVE USING NLP. The solution to the model is given in Table 13.2. It indicates that the quantity supplied and demanded equal 10, that the price is 3 (equaling the shadow price on the commodity balance row), and that consumers' plus

producers' surplus equals 25.

The above example is a simple case where we have a single supply and single demand curve. Clearly, no one would solve this problem using nonlinear programming, as it could be easily solved by hand. However, the problem does illustrate the formulation of price endogenous models.

### 13.1 Spatial Equilibrium

A common price endogenous model application involves the spatial equilibrium problem. This problem is an extension of the transportation problem relaxing the assumption of fixed supply and demand. The problem is motivated as follows. Production and/or consumption usually occurs in spatially separated regions, each of which have supply and demand relations. In a solution, if the regional prices differ by more than the interregional cost of transporting goods, then trade will occur and the price difference will be driven down to the transport cost. Modeling of this situation addresses the questions of who will produce and consume what quantities and what level of trade will occur.

Takayama and Judge (1973) developed the spatial equilibrium model to deal with such situations. Suppose that in region  $i$  the demand for the good of interest is given by

$$p_{di} = f_i(Q_{di})$$

where  $p_{di}$  is the demand price in region  $i$  while  $Q_{di}$  is the quantity demanded. Simultaneously suppose the supply function for region  $i$  is

$$p_{si} = s_i(Q_{si})$$

where  $p_{si}$  is the supply price in region  $i$ , and  $Q_{si}$  the quantity supplied. A "quasi-welfare function" for each region can be defined as the area between the supply and demand curves,

$$W_i(Q_{si}, Q_{di}) = \int_0^{Q_{di}} p_{di} dQ_{di} - \int_0^{Q_{si}} p_{si} dQ_{si}$$

The total welfare function across all regions is the sum of the welfare functions in each region less total transport costs. Suppose  $T_{ij}$  represents the amount of good shipped from  $i$  to  $j$  at cost  $c_{ij}$ . Then the net welfare is

$$NW = \sum_i W_i(Q_{di}, Q_{si}) + \sum_{i,j} c_{ij} T_{ij}$$

In turn we may form an optimization problem with the NW expression as the objective function plus the constraints from the transportation model. These constraints involve a demand balance requiring that incoming shipments to a region be greater than or equal to regional demand,

$$Q_{di} \geq \sum_j T_{ji} \text{ for all } i$$

and a supply balance requiring that outgoing shipments do not exceed regional supply

$$Q_{si} \geq \sum_j T_{ij} \text{ for all } i.$$

The resultant problem becomes

$$\begin{aligned} \text{Max} \quad & \sum_i ( \sum_o p_{di} dQ_{di} + \sum_o p_{si} dQ_{si} ) + \sum_{i,j} c_{ij} T_{ij} \\ \text{s.t.} \quad & Q_{di} - \sum_j T_{ji} \geq 0 \text{ for all } i \\ & Q_{si} - \sum_j T_{ij} \geq 0 \text{ for all } i \\ & Q_{di}, Q_{si}, T_{ij} \geq 0 \text{ for all } i \text{ and } j \end{aligned}$$

This problem yields an equilibrium solution as long as the demand curves are downward sloping and the supply curves are upward sloping. The nature of the solution and the equilibrium can best be revealed by investigating relevant parts of the Kuhn-Tucker Conditions.

$$\begin{aligned} \frac{\partial L}{\partial Q_{di}} = p_{di} + \delta_{di} & \geq 0 & \left( \frac{\partial L}{\partial Q_{di}} \right) Q_{di} & = 0 & Q_{di} & \geq 0 \\ \frac{\partial L}{\partial Q_{si}} = p_{si} + \eta_{si} & \geq 0 & \left( \frac{\partial L}{\partial Q_{si}} \right) Q_{si} & = 0 & Q_{si} & \geq 0 \\ \frac{\partial L}{\partial T_{ij}} = c_{ij} + \delta_{dj} + \eta_{si} & \geq 0 & \left( \frac{\partial L}{\partial T_{ij}} \right) T_{ij} & = 0 & T_{ij} & \geq 0 \end{aligned}$$

These conditions imply that the shadow price in region i on the first constraint set ( $\delta_{di}$ ), assuming  $Q_{di}$  is positive, equals the demand price while the second shadow price  $Q_{si}$  equals the supply price if  $Q_{si}$  is positive. The transportation activities insure that the demand price in a region must be less than the

supply prices in all other regions plus transport costs.

The solution to this problem yields the level of supply by region ( $Q_{si}$ ), the level of consumption by region ( $Q_{di}$ ), and the level of trade between regions ( $T_{ij}$ ) as well as the level of internal consumption ( $T_{ii}$ ). Price in each region is found in the dual variables.

The relationships between the equilibrium prices can take on one of several cases. Namely: a) if region  $i$  fills some of its own demand (i.e.  $T_{ii} > 0$ ), then the domestic supply and demand prices are equal; b) if region  $i$  exports to region  $j$ , ( $T_{ij} > 0$ ), then the demand price in region  $j$  equals the supply price in region  $i$  plus transport cost; c) if region  $j$  does not export to region  $i$ , then generally  $P_{dj} < P_{si} + C_{ij}$  indicates trade is not desirable since the price differential will not support the transport cost.

In this problem, the variable  $T_{ii}$  represents the quantity produced in region  $i$  and consumed in that region. For example, suppose there are 3 regions, then total supply in region 1 is denoted by  $Q_{s1}$ . Total exports to region 2 and region 3 are  $T_{12} + T_{13}$ . The amount produced in region 1 and not exported, thus locally consumed, is

$$Q_{s1} - T_{12} - T_{13} = T_{11}$$

In inequality form, the balance is

$$Q_{s1} \geq T_{11} + T_{12} + T_{13}$$

The spatial equilibrium literature commonly deals with a special case of this problem namely the case where the supply and demand functions are both linear, i.e.,

$$p_{si} = a_i + b_i Q_{si}, \text{ and } p_{di} = e_i + f_i Q_{di}$$

In this case the objective function is quadratic and becomes:

$$\text{Max } \sum_i (e_i Q_{di} - 1/2 f_i Q_{di}^2 - a_i Q_{si} - 1/2 b_i Q_{si}^2) - \sum_{i,j} c_{ij} T_{ij}$$

### Example

Suppose we have three entities (US, Europe, Japan) trading a single homogeneous commodity.

Suppose supply curves are present only in the US and Europe and the parameters of these curves are

$$p_{s,U} = 25 + Q_{s,U}$$

$$p_{s,E} = 35 + Q_{s,E}$$

while the demand curves are

$$p_{d,U} = 150 - Q_{d,U}$$

$$p_{d,E} = 155 - Q_{d,E}$$

$$p_{d,J} = 160 - Q_{d,J}$$

and internal transport is free. Also suppose transport between the US and Europe costs 3 in either direction,

while it costs 4 between the US and Japan and 5 between Europe and Japan. The formulation of this problem is

$$\begin{aligned} \text{Max } & 150Q_{d,U} - \frac{1}{2}Q_{d,U}^2 - 155Q_{d,E} - \frac{1}{2}Q_{d,E}^2 - 160Q_{d,J} - \frac{1}{2}Q_{d,J}^2 \\ & + 25Q_{s,U} - \frac{1}{2}Q_{s,U}^2 + 35Q_{s,E} - \frac{1}{2}Q_{s,E}^2 \\ & + 0T_{U,U} + 3T_{U,E} + 4T_{U,J} + 3T_{E,U} + 0T_{E,E} + 5T_{E,J} \\ & + 4T_{J,U} + 5T_{J,E} \\ \text{s.t. } & Q_{d,U} = T_{U,U} + T_{U,E} + T_{U,J} \\ & Q_{d,E} = T_{U,E} + T_{E,U} + T_{E,E} \\ & Q_{d,J} = T_{U,J} + T_{E,J} \\ & Q_{s,U} = T_{U,U} + T_{U,E} + T_{U,J} \\ & Q_{s,E} = T_{E,U} + T_{E,E} + T_{E,J} \\ & T_{U,U}, T_{U,E}, T_{U,J}, T_{E,U}, T_{E,E}, T_{E,J} \geq 0 \end{aligned}$$

The solution to this problem yields an objective function value of 9193.6. The optimal values of the variables are shown in Table 13.3.

This solution indicates consumption of 45.4 units in the U.S., and 51.4 in both Europe and Japan, while 79.6 units are supplied in the US and 68.6 in Europe. The U.S. and Europe both get all of their consumption quantities from domestic production while the U.S. exports 34.2 units to Japan and Europe exports 17.2. The equilibrium prices appear in the shadow price column. The price in the U.S. is 104.6 while the European price is 103.6. Note the Japanese price is 108.6 which is higher than the price in the other two regions by the transport cost. These prices may also be recovered by plugging the equilibrium quantities into the demand and supply curves.

The utility of this model may be demonstrated by performing some slight extensions. Suppose we use the model to examine the costs and effects of trade barriers and their cost. Specifically consider

model solution a) without any trade, b) with the U.S. imposing a quota of 2 units, and c) with the U.S. imposing a 1 unit export tax while Europe imposes a 1 unit export subsidy. Modeling these items involves the addition of a constraint which limits exports from a region to zero, the quota or an infinite amount depending on the scenario and the alteration of the inter-country transport costs to reflect the subsidy/tax. The GAMS file SPATEQ shows the implementation of these features. Table 13.4 presents a summary of the case solutions. Note that the expected results occur. Without trade domestic consumers in the U.S. and Europe receive cheaper prices and consume more, but Japanese consumers receive nothing. Simultaneously U.S. and European producers supply less and receive lower prices. Under the U.S. quota, a trade realignment occurs. This also happens under the tax and subsidy scenario. One may also interpret the objective function in terms of welfare and could decompose it to reveal the welfare positions of the consumers and producers by country. All in all, this example illustrates the potential usefulness of the spatial equilibrium, price endogenous structure.

### 13.2 Multi-Market Case

The price endogenous model may also be extended to multiple markets and products. Suppose one wishes to construct a model with multiple sources of supply and demand. For a single homogenous commodity let us illustrate this with an example.

Suppose we have two sources of wheat supply (domestic and import) and three sources of demand (wheat for making bread, wheat for making cereal, and wheat for export). Define  $Q_d$  and  $Q_i$  as the quantities supplied by domestic and import sources, respectively, with  $P_{sd}$  and  $P_{si}$  as the corresponding prices. Further, define  $X_b$ ,  $X_c$  and  $X_e$  as the quantities of bread, cereal and exported wheat demanded, with  $P_{db}$ ,  $P_{dc}$  and  $P_{de}$  the accompanying prices. Let the supply and demand curves be dependent on own price only with the supply curves:

$$\text{domestic supply: } P_{sd} \quad | \quad 2.0 \quad \% \quad 0.003 \quad Q_d,$$

$$\text{import supply: } P_{si} \quad | \quad 3.1 \quad \% \quad 0.0001 \quad Q_i.$$

and the demand curves:

$$\text{bread demand: } P_{db} \quad | \quad 0.75 \quad \& \quad 0.0004 \quad X_b,$$

$$\text{cereal demand: } P_{dc} \quad | \quad 0.80 \quad \& \quad 0.0003 \quad X_c,$$

$$\text{export demand: } P_{de} \quad | \quad 3.40 \quad \& \quad 0.0001 \quad X_e.$$



Also suppose that one bushel (bu.) of wheat yields 5 units of bread, 6 units cereal or 1 exported bushel.

A problem which depicts this situation is

$$\begin{aligned} \text{Max } & (0.75 - 1/2(.0004X_b)X_b) + (0.80 - 1/2(.0003X_c)X_c) + (3.4 - 1/2(.0001X_e)X_e) + (2.0 - 1/2(.003Q_d)Q_d) + (3.1 - 1/2(.0001Q_i)Q_i) \\ \text{s.t. } & 1/5X_b + 1/6X_c + X_e + Q_d + Q_i = 0 \\ & X, Q \geq 0 \end{aligned}$$

Readers may verify that at the optimal solution the Kuhn-Tucker conditions equate the price of wheat in the supply and demand markets as well as the quantity forming an overall equilibrium. The solution of this example arises from the file MARKETS and is given in Table 13.5. Now the question is, "What does the objective function represent?"

$$(0.75 - 1/2(.0004X_b)X_b)$$

The term is the area under the price curve for bread. Similarly, the other expressions are the integrals under the other curves. Thus, we have the integrals under the demand curves less the integrals under the supply curves leading us to a measure of the areas between the curves. The area between demand and supply functions is a measure of producers' plus consumers' surplus. Alternatively, this may be viewed as a technical behavioral objective whose purpose is to equate prices in markets.

This example again illustrates how price endogenous models can be constructed to account for multiple markets. Again, the nonlinear part of the model takes into account the price responsiveness in the demand and supply curves. This model has an explicit supply curve for the product wheat, composed of the aggregate of the two supply curves, as well as a demand curve which is the aggregate of demand for wheat in the production of three products.

### 13.3 Implicit Supply - Multiple Factors/Products

The above models involve explicit supply curves and production using a single input. However, one can depict multiple products, factors and production processes. Such models have exogenous factor supply and product demand curves, but implicit factor demand and product supply. A model of such a case is

$$\text{Max} \int_0^{Z_h} P_{dh}(Z_h) dZ_h \quad \& \quad \int_0^{X_i} P_{si}(X_i) dX_i$$

s.t. This problem assumes that a number of different types of firms (\$) are being modeled. Each firm has a finite set of production processes (k) which depict particular ways of combining fixed factors (j) with purchased factors (i) to produce commodities (h). The symbols  $P_{dh}(Z_h)$  are the inverse demand function for the h<sup>th</sup> commodity;  $Z_h$  is the quantity of commodity h that is consumed;  $P_{si}(X_i)$  is the inverse supply curve for the i<sup>th</sup> purchased input;  $X_i$  is the quantity of the i<sup>th</sup> factor supplied;  $Q_{sk}$  is the level of production process k undertaken by firm \$;  $C_{hsk}$  is the yield of output h from production process k;  $b_{jsk}$  is the quantity of the j<sup>th</sup> owned fixed factor used in producing  $Q_{sk}$ ;  $a_{isk}$  is the amount of the i<sup>th</sup> purchased factor used in producing  $Q_{sk}$  and  $Y_{js}$  is the endowment of the j<sup>th</sup> owned factor available to firm \$.

An investigation of the Kuhn-Tucker conditions would show that the shadow price on the first and second rows are respectively the demand and supply prices. The conditions for the Q variable indicates that production levels are set so the marginal value of the commodities produced is less than or equal to the marginal costs of the owned and fixed factors for each  $Q_{sk}$ .

The model formulation assumes that: 1) the supply and demand equations are integrable (we will return to this assumption later, but for now we assume path independent integrals); and, 2) product demand and factor supply functions are truly exogenous to the model (i.e., there is no income effect).

The integral of the product demand and factor supply functions makes the objective function equal consumers' plus producers' surplus or net social benefit. The solution of the model generates equilibrium price and quantity for each output, and purchased input, along with the imputed values for the owned factors of production.

The model formulation assumes that the sector is composed of many micro-units, none of which can individually influence output or factor prices. Each micro-unit supplies output at the point where marginal cost equals product price, and utilizes purchased inputs at the point where the marginal value product of each purchased input equals its market price. Thus, the sectoral supply of output schedule corresponds to an aggregate marginal cost schedule, and the sectoral derived demand schedule for purchased inputs corresponds to the aggregate marginal value product schedule. Hence, the model does

not take product supply or factor demand schedules as input, rather these schedules are derived internally based upon production possibilities, output demand and purchased input supply.

The competitive behavior simulating properties of this formulation provide a powerful tool for policy makers. Excepting centrally planned economies, the government cannot dictate production patterns consistent with its objectives. This formulation recognizes the difference and possible conflict between government and producer objectives (see Candler, Fortuny, and McCarl for elaboration). The model allows policy analysts to specify changes designed to meet some government objective, then simulate sectoral response to the policy change. The model does not assume participants respond to government "wants"; each producer optimally adjusts so as to maximize profits. Producer adjustment is endogenous to the model.

Example

Suppose we make some modifications to the block diagonal problem in Chapter 7 adding product demand and labor supply curves. Namely let us simplify the problem by only allowing sales from the first plant dropping the sales activities from the other plants. We will also specify linear product demand and labor supply curves. The curves are passed through a known price quantity point which has a particular elasticity at that point. Namely given the elasticity ( $\epsilon$ ), and known price quantity point (P, Q) then the slope (b) is found as follows. We know that slope equals

$$b_S = \frac{\Delta P}{\Delta Q}$$

while the elasticity is

$$\epsilon = \frac{\Delta Q}{\Delta P} \cdot \frac{Q}{P}$$

This implies that

$$bS = \frac{P}{Q} - \frac{P}{Q}$$

and then if

$$P = a + bQ$$

the intercept is

$$a = P - bQ.$$

In setting up the model, the assumed price, quantity and elasticity by commodity as well as the computed intercept and slope are given below

Product Sale					
Commodity	Price	Quantity	Elasticity	Computed Intercept (a)	Computed Slope (b)
Functional Chairs	82	20	-0.5	247	-8.2
Functional Tables	200	10	-0.3	867	-66.7
Functional Sets	600	30	-0.2	3600	-100
Fancy Chairs	105	5	-0.6	280	-35
Fancy Tables	300	10	-1.2	550	-25
Fancy Sets	1100	20	-0.8	2475	-68.8

Labor Supply					
Plant	Price	Quantity	Elasticity	Computed Intercept (a)	Computed Slope (b)
Plant1	20	175	1	0	.114
Plant2	20	125	1	0	.160
Plant3	20	210	1	0	.095

The resultant model is given in Table 13.6 where the objective function terms for the demand variables marked with "w" equal

$$a * Q + \frac{1}{2} b * Q^2$$

where  $Q$  depicts the quantity of the variable. The intercept and slope are as in the above table.

Similarly, those supply terms marked with "-Z", equal

$$-(\alpha * Q + \frac{1}{2} \beta * Q^2)$$

where the intercept and slope are from the labor supply table above.

The solution to this problem is given in Table 13.7 (see file ACTANAL).

Note the balances give the market prices of chairs and tables while the plant level labor balances give the labor prices. The overall objective function value again equals consumers' plus producers' surplus.

### 13.4 Aggregation

An important sector modeling topic involves aggregation. Namely, the implicit supply model assumes that there are submodels present for each firm in the sector. This is usually not practical. Such models typically deal with the aggregate response across groups of firms. Two approaches have been proposed for the formation of such an aggregate representation. The first involves derivation of conditions under which a set of models can each represent more than one entity. Such conditions require that the problems have identical constraint matrices, proportional right hand sides and objective functions (Day, 1969). The second approach involves a reformulation of the programming model. We will deal further with this reformulation here.

The reformulation approach is based upon Dantzig and Wolfe decomposition and suggestions in McCarl. Dantzig and Wolfe based their scheme on the property that the solution to a subproblem or group of subproblems will occur at the extreme points of the subproblem(s). Thus, one can reformulate the problem so that it contains the extreme point solutions from the subproblems. Formally this can be expressed as follows. Given the problem

$$\begin{aligned}
\text{Max} \quad & \int_h \int_0^{Z_h} P_{dh} ( Z_h ) dZ_h \quad \& \quad \int_i \int_0^{X_i} P_{si} ( X_i ) dX_i \\
\text{s.t.} \quad & Z_h \quad \& \quad \int_{\$k} C_{h\$k} Q_{\$k} \# \quad 0 \quad \text{for all } h \\
& \quad \& \quad X_i \quad \% \quad \int_{\$k} a_{i\$k} Q_{\$k} \# \quad 0 \quad \text{for all } i \\
& \quad \quad \quad \int_k b_{j\$k} Q_{\$k} \# \quad Y_{j\$} \quad \text{for all } j \text{ and } \$ \\
& Z_h, \quad X_i, \quad Q_{\$k} \quad \$ \quad 0 \quad \text{for all } i, h, k \text{ and } \$
\end{aligned}$$

suppose we group the firms \$ into subsets \$r\_m(\$ where \$r\_m\$ depicts the \$m^{th}\$ aggregate firm grouping. In turn,

suppose we have a set of \$s\$ feasible solutions \$Q\_{\\$k}\$ and add up their aggregate levels of production and input usage such that

$$\begin{aligned}
Z_h^{ms} &= \int_{\$r_m(\$)} \int_k C_{h\$k} Q_{\$k}^s \quad \text{for all } m, h, \text{ and } s \\
X_i^{ms} &= \int_{\$r_m(\$)} \int_k a_{i\$k} Q_{\$k}^s \quad \text{for all } m, i, \text{ and } s.
\end{aligned}$$

This in turn can be used in the aggregate problem:

$$\begin{aligned}
\text{Max} \quad & \int_h \int_0^{Z_h} P_{dh} ( Z_h ) dZ_h \quad \& \quad \int_i \int_0^{X_i} P_{si} ( X_i ) dX_i \\
\text{s.t.} \quad & Z_h \quad \& \quad \int_{ms} Z_h^{ms} \delta_{ms} \# \quad 0 \quad \text{for all } h \\
& \quad \& \quad X_i \quad \% \quad \int_{ms} X_i^{ms} \delta_{ms} \# \quad 0 \quad \text{for all } i \\
& \quad \quad \quad \int_s \delta_{ms} \# \quad 1 \quad \text{for all } m \\
& Z_h, \quad X_i, \quad \delta_{ms} \quad \$ \quad 0 \quad \text{for all } i, h, m \text{ and } s
\end{aligned}$$

This model differs in two major ways from those above. First, the firm response variables have data requirements not in terms of individual production possibilities, but rather in terms of total production

and consumption of the sector wide outputs and inputs accumulated across the firms in each group. In addition, rather than using individual resource constraints we now require a convex combination of the total output/input vectors. This will be feasible in the subproblems since any combination of two feasible subproblem solutions is feasible. Implicitly these solutions contain all the firm level resource restrictions and production possibilities coded within them.

The candidate solution vectors (i.e., the values of  $X_i^{ms}, Z_h^{ms}$ ) must be developed. These can be generated either by formally solving the linear programming subproblems for different prices or by selecting a historical set of observed feasible mixes or firms. This is discussed further in Onal and McCarl (1989, 1991).

### Example

Suppose we have a problem with four production subproblems falling into two states where the first two firms are in state 1 and the second two are in state 2. Further suppose the firms each produce two goods and use miscellaneous inputs, labor and land. Suppose the land constraint is firm specific, the labor constraint is state specific and the miscellaneous inputs constraint is national. Suppose the supply and demand curves are in Table 13.8 and the rest of the data are as given in the tableau (Table 13.9). Aggregation is introduced into this problem by considering using two state level models. Suppose over time we have observed state crop mixes as in Table 13.10. We may then reformulate the model and, rather than include all the firms and resource constraints, we simply put in the total input and output use for the observed solutions (Table 13.11). The resultant national solutions before (see file BEFORAGG) and after (see file AFTERAGG) the aggregation process are given in Table 13.12a and 13.12b. Notice that there is not a great deal of difference in these optimum solutions.

This example is indicative of a general approach to such problems. Namely, if we were trying to represent all of the farms in a sector and could obtain production and input usages by state, we could modify the model to force a convex combination of historically observed activity. This is done in the

sector models used by McCarl (1982b); Hamilton, McCarl and Adams; and Chang et al.

### 13.5 A Digression on the Assumptions

To formulate the above models or any other multi-product or multi-input model, one must assume integrability of product demand and purchased input supply functions as well as partial equilibrium. In this section, we will discuss these assumptions and suggest ways of relaxing them. Integrability requires that the Jacobians of the product demand equations and purchased input supply functions be symmetric (Hurwicz and Uzawa). The system of product demand functions is

$$P' = G \text{ \& \ } HZ$$

and the system of purchased input supply functions is

$$R' = E \text{ \% \ } FX$$

The Jacobians of the demand and supply equations are H and F, respectively. Symmetry of H and F implies that cross price effects across all commodity pairs are equal; i.e.,

$$\begin{aligned} \frac{\partial P_{dr}}{\partial Q_{dh}} &= \frac{\partial P_{dh}}{\partial Q_{dr}} \text{ for all } r \neq h \\ \frac{\partial P_{sr}}{\partial Q_{sh}} &= \frac{\partial P_{sh}}{\partial Q_{sr}} \text{ for all } r \neq h \end{aligned}$$

In the case of supply functions, classical production theory assumptions yield the symmetry conditions. The Slutsky decomposition reveals that for the demand functions, the cross price derivatives consist of a symmetric substitution effect and an income effect. The integrability assumption requires the income



effect to be identical across all pairs of commodities or to be zero.

Some authors argue that there need be no concern regarding symmetry. Since the objective function is a quadratic form. Then, given any square matrix, H, a quadratic form is the scalar quantity that results when H is pre- and post- multiplied by a conformable vector,

$$v = x' H A x$$

where v is the value of the quadratic form. Mathematically, if we replace H with the symmetric matrix B

$$B = 1/2 (H + H')$$

One can easily show that

$$x' B x = x' H A x$$

Thus, if H is not symmetric, it can be replaced by B, and the value of the objective function remains unchanged. But, when the first order conditions are formed, the derivatives are altered. In particular if one integrates the above demand curve, we get

$$\frac{\partial (G + 1/2 Z' H Z)}{\partial Z} = \frac{\partial G}{\partial Z} + \frac{\partial (1/2 Z' H Z)}{\partial Z} = G + 1/2 Z' (H + H')$$

which would not give the demand price. Thus marginal cost and product price are no longer equilibrated.

Models can be formed which can handle asymmetry. Price and quantity variables can be included in the primal model (Plessner and Heady). Thus, both price and quantity equilibrium conditions are imposed on the primal problem, as contrasted with the above specification in which only quantity equilibrium conditions are imposed on the primal, and price equilibrium conditions are found in the dual. Another approach is linear complementarity programming (Takayama and Judge; Stoecker; or Polito). In this case, the objective function no longer represents consumers' plus producers' surplus. For further discussion, see Takayama and Judge or Martin.

The partial equilibrium assumption arises because the formulation does not incorporate the income generated by the sector as a simultaneous shifter of demand for products included in the model.

If the entity modeled is small relative to the entire economy, this should not be a problem. If a major proportion of consumers included in the model are also producers, then the model inadequately describes the linkages in the economy. A formulation which does not require the partial equilibrium assumption was developed by Yaron, who specified a lagged relationship in which aggregate consumer demand in the current period is a function of income in the previous period. Norton and Scandizzo have relaxed this assumption in a simultaneous fashion in which demand is specified as a function of current consumer income. Integrability is a consequence as an income shifter is explicitly introduced, leaving only the symmetric substitution terms.

For further discussion of empirical specification of price endogenous models, see the review papers by McCarl and Spreen or Norton and Schiefer.

### 13.6 Imperfect Competition

So far, we have basically dealt with price endogeneity starting from Samuelson's approach, casting a set of first-order conditions and discovering the QP that would yield such a set. Another approach, however, can be taken. Suppose one begins with a classic LP problem involving two goods and a single constraint; i.e.,

$$\begin{aligned} \text{Max } & P_1 X \text{ \& } P_2 Q \\ \text{s.t. } & X \text{ \& } Q \# 0 \\ & X, Q \$ 0 \end{aligned}$$

However, rather than  $P_1$  and  $P_2$  being fixed, suppose that we assume that they are functionally dependent upon quantity as given by

$$\begin{aligned} P_1 & ' a \text{ \& } bX \\ P_2 & ' c \% dQ \end{aligned}$$

Now suppose one simply substitutes for  $P_1$  and  $P_2$  in the objective function. This yields the problem

$$\begin{aligned} \text{Max} \quad & aX + bX^2 + cQ + dQ^2 \\ \text{s.t.} \quad & X + Q = 0 \\ & X, Q \geq 0 \end{aligned}$$

Note the absence of the  $1/2$ 's in the objective function. If one applies Kuhn-Tucker conditions to this problem, the conditions on the  $X$  variables, assuming they take on non-zero levels, are

$$\begin{aligned} a + 2bX + \lambda &= 0 \\ c + 2dQ + \lambda &= 0 \end{aligned}$$

The solution to this set of equations implies that the dual variable ( $\lambda$ ) is equated to something with twice the slope of the demand curve. Readers familiar with the imperfect competition literature will recognize this as an equation of marginal revenue with marginal cost. Such actions are only consistent with the behavior of perfectly discriminating monopolists - monopsonists. This indicates a couple of things about the approach to price equilibrating models: if one is not careful and does not put the integrals in, one simulates imperfect competition. In fact, there are four cases involving the integrals ( $1/2$ 's in the quadratic case). Given the supply and demand relationships, one may model as follows

[I] Monopolist & Monopsonist

$$\begin{aligned} \text{Max} \quad & X(a + bX) + Q(C + dQ) \\ & X + Q = 0 \end{aligned}$$

[II] Monopolist & Supply Competitor

$$\begin{aligned} \text{Max} \quad & X(a + bX) + Q(C + 1/2dQ) \\ & X + Q = 0 \end{aligned}$$

[III] Demand Competitor & Monopsonist

$$\begin{aligned} \text{Max} \quad & X(a + 1/2bX) + Q(C + dQ) \\ & X + Q = 0 \end{aligned}$$

[IV] Competitor in Both Markets

$$\begin{aligned} \text{Max} \quad & X(a + 1/2bX) + Q(C + 1/2dQ) \\ & X + Q = 0 \end{aligned}$$

The solutions to these problems are graphed in Figure 13.3. Using the wheat problem, the numerical

solutions shown in Table 13.13 are determined under the four alternative behavioral assumptions. This shows that one can obtain alternative forms of competition by selectively omitting or including integrals.

Nelson and McCarl provide a more general discussion of the topic of imperfect competition under the quadratic case. They show that in each of the demand and supply curves, if the term

$$\frac{n - 1}{2n}$$

is substituted for the  $\frac{1}{2}$ , then one obtains a simulation of the effect of  $n$  firms discriminating against the demand or supply curves to this parameter is supplied. This particular term reduces to  $\frac{1}{2}$  when  $n$  approaches  $\infty$ , and 1 when  $n=1$ . Thus, it covers both the monopolistic and perfectly competitive cases. But also, for example, when  $n=2$ , the equation says to use a  $\frac{3}{4}$  to reflect two firms acting under imperfect competition against a particular supply curve. Readers should be careful in using this formulation, as it indicates how one discriminates against the entity which the particular supply or demand curve depicts, not how that entity discriminates against others. Nelson and McCarl present a more careful discussion on handling other forms of imperfect competition.

### 13.7 Conclusion

In the preceding sections, price endogenous models have been developed for spatial equilibrium, multi-market, multi-product, multi-factor models, aggregate, and imperfect competition. It should be clear that these models may be combined with our earlier formulations. For example, Spreen et al. integrated a multi-product industry formulation with a disequilibrium known life type formulation in a study of the livestock sector in Guyana.

These types of models have been used in many studies, as listed in the review book by Judge and Takayama, the review papers by McCarl and Spreen, Martin, and Norton and Schiefer.

## References

- Candler, W.V., J. Fortuny-Amat, and B.A. McCarl. "The Potential Role of Multilevel Programming in Agricultural Economics." American Journal of Agricultural Economics. 63(1981):521-31.
- Chang, C.C., B.A. McCarl, J. Mjelde, and J.W. Richardson. "Sectoral Implications of Farm Program Modifications." American Journal of Agricultural Economics. 74(1992):38-49.
- Dantzig, G.B. and P. Wolfe. "The Decomposition Algorithm for Linear Programs." Econometrica. 29(1961):767-778.
- Day, R.H. "Exact Aggregation with Linear Programming Models -- A Note on the Sufficient Conditions Proposed by R.H. Day: Reply." American Journal of Agricultural Economics. 51(1969):686-688.
- Enke, S. "Equilibrium Among Spatially Separated Markets: Solution by Electric Analogue." Econometrica. 19(1951):40-47.
- Duloy, J.H. and R.D. Norton. "Prices and Incomes in Linear Programming Models." American Journal of Agricultural Economics. 57(1975):591-600.
- Hamilton, S.A., B.A. McCarl, and R.M. Adams. "The Effect of Aggregate Response Assumptions on Environmental Impact Analysis." American Journal of Agricultural Economics. 67,2(1985):407-413.
- Hurwicz, L. and H. Uzawa. "On the Integrability of Demand Functions." In Preferences, Utility and Demand: A Minnesota Symposium, ed. J.S. Chipman, L. Hurwicz, M.K. Richter and H. Sonnenschein, p.114. New York: Harcourt, Brace and Co., 1971.
- Judge, G.G. and T. Takayama. Studies in Economic Planning over Space and Time. Amsterdam: North-Holland Publishing Co., 1973.
- Lemke, C.E. "Bimatrix Equilibrium Points and Mathematical Programming." Management Science. 11(1965):681-689.
- Martin, L.J. "Quadratic Single and Multi-Commodity Models of Spatial Equilibrium: A Simplified Exposition." Canadian Journal of Agricultural Economics. 29(1981):21-48.
- McCarl, B.A. "Cropping Activities in Agricultural Sector Models; A Methodological Proposal." American Journal of Agricultural Economics. 64(1982):768-772.
- McCarl, B.A., H. Moskowitz and H. Furtan. "Quadratic Programming Applications" OMECA: The International Journal of Management Science. 5(1977):43-55.
- McCarl, B.A. and T.H. Spreen. "Price Endogenous Mathematical Programming as a Tool for Policy Analysis." American Journal of Agricultural Economics. 62(1980):87-102.
- Nelson, C.H. and B.A. McCarl. "Including Imperfect Competition in Spatial Equilibrium Models." Canadian Journal of Agricultural Economics. 32(1984):55-70.

- Norton, R.D. and P.L. Scandizzo. "Market Equilibrium Computations in Activity Analysis Models." Draft, International Bank for Reconstruction and Development, Washington D.C., April, 1979.
- Norton, R.D. and G.W. Schiefer. "Agricultural Sector Programming Models: A Review." European Review of Agricultural Economics. 7(1980):229-64.
- Onal, H. and B.A. McCarl. "Aggregation of Heterogeneous Firms In Mathematical Programming Models." European Journal of Agricultural Economics. 16,4(1989):499-513.
- Onal, H. and B.A. McCarl. "Exact Aggregation in Mathematical Programming Sector Models." Canadian Journal of Agricultural Economics. 39(1991):319-334.
- Plessner, Y. and E.O. Heady. "Competitive Equilibrium Solutions with Quadratic Programming." Metroeconomica. 17(1965):117-30.
- Polito, J. "Distribution Systems Planning in a Price Responsive Environment." M.S. Thesis, Purdue University, August 1977.
- Samuelson, P.A. "Spatial Price Equilibrium and Linear Programming." American Economic Review. 42(1952):283-303.
- Spreen, T.H., B.A. McCarl, and T.K. White. "Modeling the Cattle Subsector: A Demonstration in Guyana." Canadian Journal Agricultural Economics. 28(1980):17-32.
- Stoecker, A. "A Quadratic Programming Model of United States Agriculture in 1980." Ph.D. Thesis, Iowa State University, 1974.
- Takayama T. and G.G. Judge. Spatial and Temporal Price and Allocation Models. Amsterdam: North Holland Publishing Co., 1971.
- Yaron, D. "Incorporation of Income Effects in Mathematical Programming Models." Metroeconomica. 19(1967):141-60.

**Table 13.1. GAMS Formulation of Simple Price Endogenous Example**

---

```
2
4  OPTION LIMCOL = 0;
5  OPTION LIMROW = 0;
6
7  SETS      CURVEPARM  CURVE PARAMETERS  /INTERCEPT,SLOPE/
8           CURVES     TYPES OF CURVES   /DEMAND,SUPPLY/
9
10 TABLE   DATA(CURVES,CURVEPARM) SUPPLY DEMAND DATA
11
12           INTERCEPT  SLOPE
13  DEMAND      6         -0.30
14  SUPPLY      1         0.20
15
16 PARAMETERS SIGN(CURVES) SIGN ON CURVES IN OBJECTIVE FUNCTION
17           /SUPPLY -1, DEMAND 1/
18
19 POSITIVE VARIABLES  QUANTITY(CURVES) ACTIVITY LEVEL
20
21 VARIABLES           OBJ              NUMBER TO BE MAXIMIZED
22
23 EQUATIONS           OBJJ              OBJECTIVE FUNCTION
24                   BALANCE            COMMODITY BALANCE;
25
26 OBJJ..  OBJ =E= SUM(CURVES, SIGN(CURVES)*
27               (DATA(CURVES,"INTERCEPT")*QUANTITY(CURVES)
28               +0.5*DATA(CURVES,"SLOPE")*QUANTITY(CURVES)**2)) ;
29
30 BALANCE..  SUM(CURVES, SIGN(CURVES)*QUANTITY(CURVES)) =L= 0 ;
31
32 MODEL PRICEEND /ALL/ ;
33
34 SOLVE PRICEEND USING NLP MAXIMIZING OBJ ;
35
```

---

**Table 13.2. Solution to Simple Price Endogenous Model**

Variables	Level	Reduced Cost	Equation	Slack	Shadow Price
$Q_d$	10	0	Objective function	0	-1
$Q_s$	10	0	Commodity Balance	0	3

**Table 13.3. Solution to Spatial Equilibrium Model**

Objective function = 9193.6					
Variables	Value	Reduced Cost	Equation	Level	Shadow Price
Supply			Supply Balance		
U.S.	79.6	0	U.S.	0	104.6
Europe	68.6	0	Europe	0	103.6
Demand			Demand Balance		
U.S.	45.4	0	U.S.	0	104.6
Europe	51.4	0	Europe	0	103.6
Japan	51.4	0	Japan	0	108.6
Shipments					
U.S. to U.S.	45.4	0			
U.S. to Europe	0	-4			
U.S. to Japan	34.2	0			
Europe to U.S.	0	-2			
Europe to Europe	51.4	0			
Europe to Japan	17.2	0			



**Table 13.4. Solutions to Alternative Configurations of Spatial Equilibrium Model**

	Undistorted	No Trade	Scenario Quota	Tax/Subsidy
Objective	9193.6	7506.3	8761.6	9178.6
U.S. Demand	45.4	62.5	61.5	46.4
U.S. Supply	79.6	62.5	63.5	78.6
U.S. Price	104.6	87.5	88.5	103.6
Europe Demand	51.4	60	40.7	50.4
Europe Supply	68.6	60	79.3	69.6
Europe Price	103.6	95	114.3	104.6
Japan Demand	51.4	0	40.7	51.4
Japan Price	108.6	160	119.3	108.6

**Table 13.5. Solution to the Wheat Multiple Market Example**

$X_b$	255.44
$X_c$	867.15
$X_e$	1608.72
$Q_d$	413.04
$Q_i$	1391.29
$P_{db}$	0.648
$P_{dc}$	0.540
$P_{de}$	3.239
$P_{sd}$	3.239
$P_{si}$	3.239
Shadow Price	3.239

**Table 13.6. Formulation of the Implicit Supply Example**

		PLANT 1					PLANT 2					PLANT 3				
Objective	Sell Sets FC FY	Make Table FC FY	Sell Table FC	Sell Chair FC FY	Transport Chair	Make Functional Chairs	Make Fancy Chairs Norm MxSm MxLg	Labor Supply	Transport Table	Transport Chair FC FY	Make Table FC FY	Make Functional Chairs	Make Fancy Chairs Norm MxSm MxLg	Labor Supply	SHS	
P	W W		W W	W W	-Z	-5 -5	-15 -16 -16	-25 -25 -26	-Z	-7	-7 -7	-80 -	-15 -16 -16.5	-25 -25.7 -26.6	-Z	
L	1	-1	1							-1					f 0	
A	1	-1	1							-1					f 0	
N	4			1		-1					-1				f 0	
T	6			1		-1									f 0	
		3	5												f 0	
		1	1												f 50	
1															f 0	
P L						1	-1 -1 -1								f 0	
A						1									f 0	
N								-1 -1 -1							f 0	
T								1.2 1.7 0.5							f 140	
								0.8 1.3 0.2							f 90	
								0.5 0.2 1.3							f 120	
								0.4 0.4 0.4							f 0	
2								1 1.05 1.1	-1						f 0	
								0.8 0.82 0.84							f 0	
P									1						f 0	
L															f 0	
A									1						f 0	
N										1					f 0	
T											1				f 0	
															f 0	
3															f 130	
															f 100	
															f 110	
															f 0	
															f -40	

**Table 13.7. Solution of the Implicit Supply Example**

Rows				Shadow Price	Columns Variable Names	Level	Reduced Cost
Objective				95779.1			
PLANT 1	Table	FC	0	165.1	Sell FC Set	30.9	0
	Table	FY	0	228.5	Sell FY Set	23.0	0
	Chair	FC	0	85.6	Sell FC Tables	10.5	0
	Chair	FY	0	110.8	Sell FY Tables	12.9	0
	Labor		0	21.7	Sell FC Chairs	19.6	0
	Labor		0	21.7	Sell FY Chairs	4.8	0
	Top Capacity		0	20.0	Make Table FC	30.1	0
PLANT 2	Chair	FC	0	80.6	Make Table FY	20.0	0
	Inventory	FY	0	105.8	Hire Labor	189.9	0
	Small Lathe		0	35.6			
	Large Lathe		0	28.0			
	Carver		29.02	0			
	Labor		0	23.1			
	Labor		0	23.1			
PLANT 3	Table	FC	0	145.1	Transport FC Chair	105.0	0
	Inventory	FY	0	208.5	Transport FY Chair	48.9	0
	Chair	FC	0	78.6	Make Table FC	0	-69.3
	Inventory	FY	0	103.8	Make Table FY	0	-115.5
	Small Lathe		0	35.1	Make FC Chair	N	105.1
	Large Lathe		0	27.6		S	0
	Carver		0.80	0		L	0
	Labor		0	21.7	Make FY Chair	N	44.9
	Labor		0	21.7		S	0
	Top Capacity		12.69	0		L	4.0
				Hire Labor		144.4	0
PLANT 3					Transport FC Table	11.4	0
					Transport FY Table	15.9	0
					Transport FC Chair	38.2	0
					Transport FY Chair	93.9	0
					Make FC Table	11.4	0
					Make FY Table	15.9	0
					Make FC Chair	N	38.2
						S	0
						L	0
					Make FY Chair	N	75.0
						S	0
						L	19.0
					Hire Labor		227.9

**Table 13.8. Demand and Supply Parameters for Aggregation Example**

	Price	Quantity	Elasticity
Product Demands			
Cotton	225	3326	-1.5
Corn	2.10	1087	-1.1
Hired Labor Supply			
State 1	5	78.7	0.5
State 2	4.5	68.1	1.2



**Table 13.10. Crop Mix Data for use in Aggregation Example**

Region	Farm	Mix 1		Mix 2	
		Corn	Cotton	Corn	Cotton
State 1	Farm1	.3	.7	.5	.5
	Farm 2	.1	.9	.3	.7
State 2	Farm1	.6	.4	.75	.25
	Farm 2	.55	.45	.6	.4

**Table 13.11. Aggregation Example after Aggregation**

	Sales corn	Hired Labor		Crop Mixes				Misc Inputs
		State 1	State 2	State 1		State 2		
				Mix 1	Mix 2	Mix 1	Mix 2	
Obj. Func.	a	-b	-b					-1
Misc Inputs				2455	2041	1725	1445	-1
Labor State 1		-1		119.2	110.6			#
Labor State 2			-1			97.6	87.7	#
Product Corn	1			-288	-576	-9.6	-6.9	#
Balance Cotton				-18.6	-13.9	-698	-833	#
Convexity State1				1	1			#
Convexity State2						1	1	#

**Table 13.12. Solutions of Aggregation Example A Before Aggregation**

Rows	Slack	Shadow Price	Variable	Level	Reduced Cost
Objt	7777.4				
Misc Inputs	0	-1.000	Sales Cotton	31.6	0
			Sales Corn	967.4	0
State Labor State 1	0	2.318			
State Labor State 2	0	4.288	Hired Labor State 1	57.6	0
			Hired Labor State 2	64.3	0
Farm Labor 1	0	1.16			
Farm Labor 2	0	1.16	Hire Labor Farm 1, State 1	35.8	0
Farm Labor 3	0	2.14	Hire Labor Farm 2, State 2	21.8	0
Farm Labor 4	0	2.14	Hire Labor Farm 3, State 3	55.5	0
			Hire Labor Farm 4, State 4	8.8	0
Cotton	0	232.6			
Corn	0	2.31	Corn, Farm 1	0	-6.2
			Cotton, Farm 1	6.0	0
Land Farm 1	0	191.4	Corn, Farm 2	2.9	0
Land Farm 2	0	305.9	Cotton, Farm 2	1.1	0
Land Farm 3	0	230.8	Corn, Farm 3	0	-20.4
Land Farm 4	0	255.9	Cotton, Farm 3	5.0	0
			Corn, Farm 4	3.0	0
			Cotton, Farm 4	0	-5.0
			Misc Inputs	4799	0

**B After Aggregation**

Rows	Slack	Shadow Price	Variable	Level	Reduced Cost
Obj. Func.	7052.2		Sales Corn	28.2	0
Cost	0	1	Sales Cotton	985.5	0
Labor State 1	0	10.1	Hire Labor State 1	119.2	0
Labor State 2	0	6.1	Hire Labor State 2	97.5	0
Product Bal. Corn	0	247.9	Crop State 1 Mix 1	1	0
Product Bal. Cotton	0	2.28	Crop State 1 Mix 2	0	-12.9
Convex State 1	1	1603.4	Crop State 2 Mix 1	1	0
Convex State 2	1	1641.5	Crop State 2 Mix 2	0	-21.6
			Cost	4180	0

**Table 13.13. Alternative Solutions to Wheat Multiple Markets  
Example under Varying Competitive Assumptions**

	I	II	III	IV
$X_b$	127.718	142.335	226.067	255.46
$X_c$	433.579	449.821	834.526	867.16
$X_e$	804.357	1096.705	1021.340	1608.71
$Q_d$	206.521	393.533	216.311	413.04
$Q_i$	695.642	806.589	989.330	1391.29
$P_{db}$	0.699	0.693	0.660	0.649
$P_{dc}$	0.669	0.665	0.550	0.540
$P_{de}$	3.3196	3.29033	3.2979	3.239
$P_{sd}$	2.6196	3.18066	2.6489	3.239
$P_{si}$	3.6196	3.18066	3.1999	3.239
Shadow Price	3.2391	3.18066	3.2979	3.239



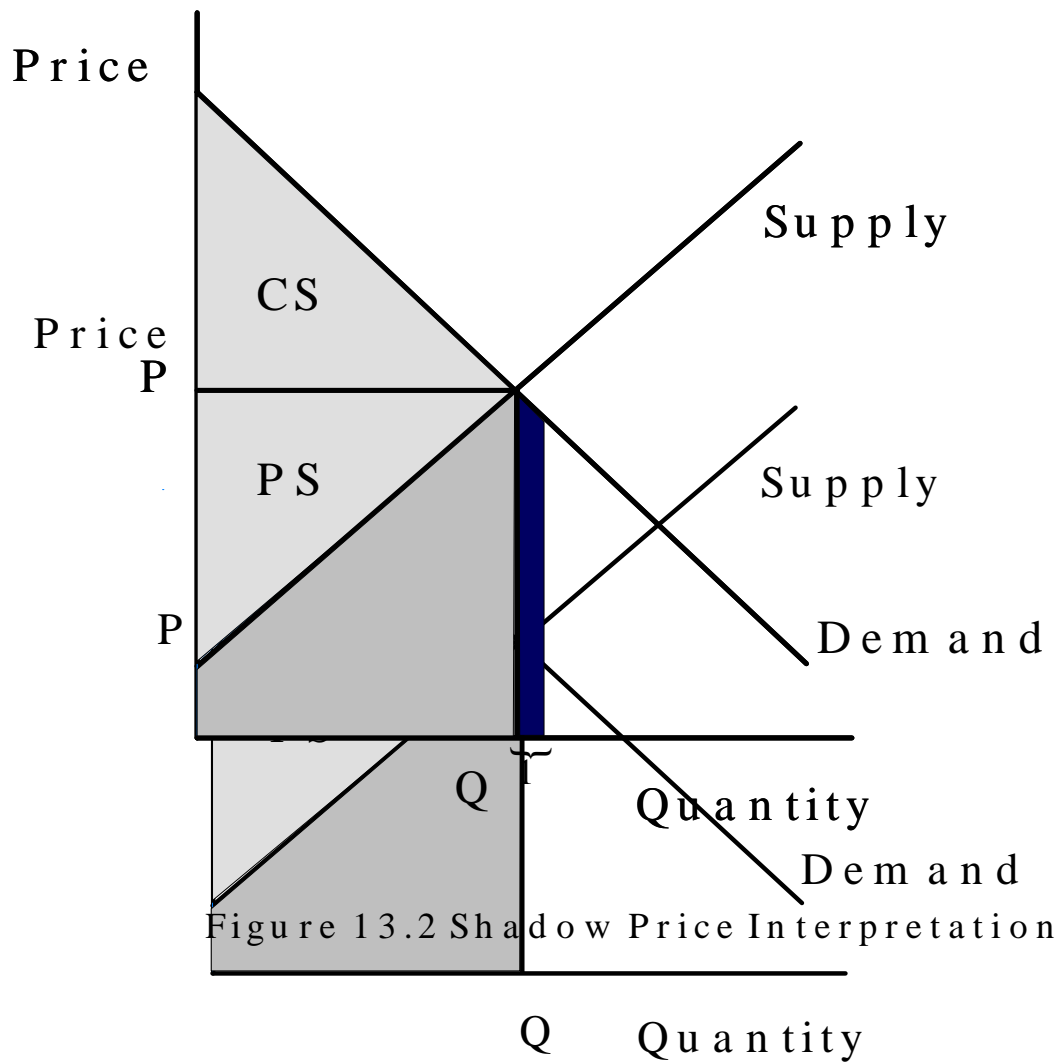


Figure 13.1 Objective Function Representation

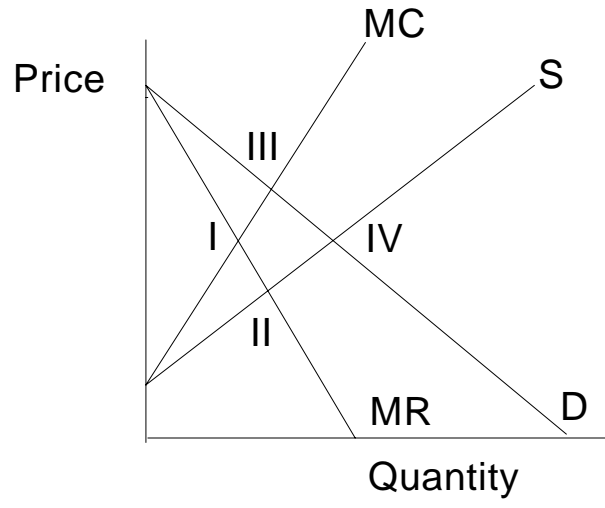


Figure 13.3 Graph of Imperfect Competition Solutions

## CHAPTER XIV: RISK MODELING

Risk is often cited as a factor which influences decisions. This chapter reviews methods for incorporating risk and risk reactions into mathematical programming models.<sup>7</sup>

Mathematical programming risk models depict the risk inherent in model parameters. Risk considerations are usually incorporated assuming that the parameter probability distribution (i.e., the risk) is known with certainty.<sup>8</sup> Usually, the task becomes one of adequately representing these distributions as well as the decision makers response to parameter risk.

The question arises: Why model risk, why not just solve the model under all combinations of the risky parameters and use the resultant plans? Such an approach is tempting, yet suffers from problems of dimensionality and certainty. The dimensionality problem is manifest in the number of possible plans; (i.e., five possible values for each of three parameters would lead to  $3^5 = 243$  possible parameter specifications). Often, there are more possible states of nature than can practically be enumerated. Furthermore, these enumerated plans suffer from a certainty problem. Every LP parameter is assumed known with perfect knowledge. Consequently, solutions reflect "certain" knowledge of the parameter values imposed. Thus, when one solves many models one gets many plans and the question remains which plan should be used.

Usually, it is desirable to generate a robust solution which yields satisfactory results across the distribution of parameter values. The risk modeling techniques discussed below are designed to yield such a plan. The "optimal" plan for a risk model generally does not place the decision maker in the best possible position for all (or maybe even any) possible events, but rather establishes a robust position across the set of possible events.

### 14.1 Decision Making and Recourse

Many different programming formulations have been posed for risk problems. An important assumption involves the potential decision maker reaction to information. The most fundamental distinction is between cases where:

- 1) all decisions must be made now with the uncertain outcomes resolved later, after all random draws from the distribution have been taken, and
- 2) some decisions are made now, then later some uncertainties are resolved followed by other decisions yet later.

These two settings are illustrated as follows. In the first case, all decisions are made then events

---

<sup>7</sup> The risk modeling problem is a form of the multiple objective programming problem so that there are parallels between the material here and that in the multi-objective chapter.

<sup>8</sup> It should be noted that risk and uncertainty are used interchangeably. Any time we discuss risk or uncertainty we assume that the probability distribution is known.

occur and outcomes are realized. This is akin to a situation where one invests now and then discovers the returns to the investment at year end without any intermediate buying or selling decisions. In the second case, one makes some decisions now, gets some information and makes subsequent decisions. Thus, one might invest at the beginning of the year, but could sell and buy during the year depending on changes in stock prices.

The main distinction is that under the first situation decisions are made before any uncertainty is resolved and no decisions are made after any of the uncertainty is resolved. In the second situation, decisions are made sequentially with some decisions made conditional upon outcomes that were subject to a probability distribution at the beginning of the time period.

These two frameworks lead to two very different types of risk programming models. The first type of model is most common and is generally called a stochastic programming model. The second type of model was originally developed by Dantzig in the early 50's and falls into the class of stochastic programming with recourse models. These approaches are discussed separately, although many

stochastic programming techniques can be used when dealing with stochastic programming with recourse problems.

#### **14.2 An Aside: Discounting Coefficients**

Before discussing formal modeling approaches, first let us consider a common, simple approach to risk used in virtually all "risk free" linear programming studies. Suppose a parameter is distributed according to some probability distribution, then a naive risk specification would simply use the mean. However, one could also use conservative price estimates (i.e., a price that one feels will be exceeded 80% of the time).

This reveals a common approach to risk. Namely, data for LP models are virtually never certain. Conservative estimates are frequently used, in turn producing conservative plans (see McCarl et al., for an example of treatment of time available). Objective function revenue coefficients may be deflated while cost coefficients are inflated. Technical coefficients and right hand sides may be treated similarly. The main difficulty with a conservative estimate based approach is the resultant probability of the solution. Conservative estimates for all parameters can imply an extremely unlikely event and an overly conservative choice of the decision variables.

#### **14.3 Stochastic Programming without Recourse**

Stochastic programming techniques generally treat risk in the objective function coefficients, technical coefficients or right hand sides separately or collectively.

##### **14.3.1 Objective Function Coefficient Risk**

Several objective function coefficient risk models have been posed. This section reviews these. First, however, some statistical background on distributions of linear sums is necessary.

Given a linear objective function

$$Z = c_1 X_1 + c_2 X_2$$

where  $X_1, X_2$  are decision variables and  $c_1, c_2$  are uncertain parameters distributed with means  $\bar{c}_1$  and  $\bar{c}_2$  as well as variances  $s_{11}, s_{22}$ , and covariance  $s_{12}$ ; then  $Z$  is distributed with mean

$$\bar{Z} = \bar{c}_1 X_1 + \bar{c}_2 X_2$$

and variance

$$F_Z^2 = s_{11} X_1^2 + s_{22} X_2^2 + 2 s_{21} X_1 X_2.$$

In matrix terms the mean and variance of  $Z$  are

$$(\bar{C}X, X^T S X)$$

where in the two by two case

$$\bar{C} = [\bar{c}_1 \quad \bar{c}_2] \quad S = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix}.$$

Defining terms

$s_{ii}$  is the variance of the objective function coefficient of  $X_i$ , which is calculated using the formula  $s_{ik} = \frac{1}{N} \sum (c_{ik} - \bar{c}_i)^2$  where  $c_{ik}$  is the  $k^{\text{th}}$  observation on the objective value of  $X_i$  and  $N$  is the number of observations.

$s_{ij}$  for  $i \neq j$  is the covariance of the objective function coefficients between  $X_i$  and  $X_j$ , calculated by the formula  $s_{ij} = \frac{1}{N} \sum (c_{ik} - \bar{c}_i)(c_{jk} - \bar{c}_j)$ . Note  $s_{ij} = s_{ji}$ .

$\bar{c}_i$  is the mean value of the objective function coefficient associated with  $X_i$ , calculated by  $\bar{c}_i = \frac{1}{N} \sum c_{ik}$ . (Assuming an equally likely probability of occurrence.)

#### 14.3.1.1 Mean-Variance Analysis

The above expressions define the mean and variance of a LP objective function with risky  $c$  parameters. Markowitz exploited this in the original mean-variance portfolio choice formulation.

The portfolio choice problem involves development of an "optimal" investment strategy. The variables indicate the amount of funds invested in each risky investment subject to a total funds constraint. Markowitz motivated the formulation by observing that investors only place a portion, not all, of their funds in the highest-yielding investment. This, he argued, indicated that a LP formulation is inappropriate since such an LP would reflect investment of all funds in the highest yielding alternative (since there is a single constraint). This divergence between observed and modeled behavior led Markowitz to include a

---

<sup>9</sup> One could also use the divisor  $N-1$  when working with a sample.

variance term resulting in the so-called expected value variance (E-V) model.

Freund (1956) developed a related model, apparently independently, which has become the most commonly used E-V model. The portfolio context of his formulation is

$$\begin{aligned} \text{Max } & \sum_j \bar{c}_j X_j \quad \& \quad M \sum_{j,k} s_{jk} X_j X_k \\ \text{s.t. } & \sum_j X_j = 1 \\ & X_j \geq 0 \quad \text{for all } j \end{aligned}$$

Here the objective function maximizes expected income ( $\bar{c}X$ ) less a "risk aversion coefficient" ( $M$ ) times the variance of total income ( $X'SX$ ). The model assumes that decision makers will trade expected income for reduced variance.

In this context Markowitz discussed the E-V efficient frontier which is the locus of points exhibiting minimum variance for a given expected income, and/or maximum expected income for a given variance of income (Figure 14.1 gives the frontier for the example below). Such points are efficient for a decision maker with positive preference for income, negative preference for variance and indifference to other factors.

The E-V problem can handle problem contexts broader than the portfolio example. A general formulation in the resource allocation context is

$$\begin{aligned} \text{Max } & \bar{C}X \quad \& \quad M X'SX \\ \text{s.t. } & AX = b \\ & X \geq 0 \end{aligned}$$

where  $\bar{C}$  is average returns from producing  $X$  and  $S$  gives the associated variance-covariance matrix.

#### 14.3.1.1.1 Example

Assume an investor wishes to develop a stock portfolio given the stock annual returns information shown in Table 14.1, 500 dollars to invest and prices of stock one \$22.00, stock two \$30.00, stock three \$28.00 and stock four \$26.00.

The first stage in model application is to compute average returns and the variance-covariance matrix of total net returns. The mean returns and variance - covariance matrix are shown in Table 14.2. In turn the objective function is

$$\text{Max } [ 4.70 \quad 7.60 \quad 8.30 \quad 5.80 ] \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} \quad \& \quad M [X_1 \quad X_2 \quad X_3 \quad X_4] \begin{bmatrix} \%3.21 & \& 3.52 & \% 6.99 & \%0.04 \\ \& 3.52 & \% 5.84 & \& 13.68 & \%0.12 \\ \%6.99 & \& 13.68 & \%61.81 & \& 1.64 \\ \%0.04 & \% 0.12 & \& 1.64 & \%0.36 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix}$$

or, in scalar notation

$$\begin{array}{l} \text{Max} \quad 4.70 X_1 \quad \% \quad 7.60 X_2 \quad \% \quad 8.30 X_3 \quad \% \quad 5.80 X_4 \\ \\ \& \quad M \quad \left( \begin{array}{l} \% \quad 3.21 X_1^2 \quad \& \quad 3.52 X_1 X_2 \quad \% \quad 6.99 X_1 X_3 \quad \% \quad 0.04 X_1 X_4 \\ \& \quad 3.52 X_2 X_1 \quad \% \quad 5.84 X_2^2 \quad \& \quad 13.68 X_2 X_3 \quad \% \quad 0.12 X_2 X_4 \\ \% \quad 6.99 X_3 X_1 \quad \& \quad 13.68 X_3 X_2 \quad \% \quad 61.81 X_3^2 \quad \& \quad 1.64 X_3 X_4 \\ \% \quad 0.04 X_4 X_1 \quad \% \quad 0.12 X_4 X_2 \quad \& \quad 1.64 X_4 X_3 \quad \% \quad 0.36 X_4^2 \end{array} \right) \end{array}$$

This objective function is maximized subject to a constraint on investable funds:

$$22X_1 \% 30X_2 \% 28X_3 \% 26X_4 \# 500$$

and non-negativity conditions on the variables.

Empirically, this problem is solved for various M values as implemented in the GAMS instructions in Table 14.3 or in the EVPORTFO file. The solutions, at selected values of M, are shown in Table 14.4, while Figure 14.1 gives the efficient frontier.

The model yields the profit maximizing solution ( $X_1=X_2=X_4=0, X_3=17.86$ ) for low risk aversion parameters ( $M < 0.0005$ ). As the risk aversion parameter increases, then  $X_2$  comes into the solution. The simultaneous use of  $X_2$  and  $X_3$  coupled with their negative covariance reduces the variance of total returns. This pattern continues as M increases. For example, when M equals 0.012 expected returns have fallen by \$17 or 11%, while the standard deviation of total returns has fallen by \$117 or 80%. For yet higher values of the risk aversion parameter, investment in  $X_1$  increases, then later  $X_4$  is added.

Three other aspects of these results are worth noting. First, the shadow price on investable capital continually decreases as the risk aversion parameter (M) increases. This reflects an increasing risk discount as risk aversion increases. Second, solutions are reported only for selected values of M. However, any change in M leads to a different solution and an infinite number of alternative M's are possible; e.g., all solutions between M values of 0.0005 and 0.0075 are convex combinations of those two solutions. Third, when M becomes sufficiently large, the model does not use all its resources. In this particular case, when M exceeds 2.5, not all funds are invested.

#### 14.3.1.1.2 Markowitz's E-V Formulation

Markowitz's original formulation of the E-V problem minimized variance subject to a given level of expected income as in the multi-objective programming lexicographic formulation.

Algebraically, this model is

$$\begin{aligned} \text{Min } & X^T S X \\ \text{s.t. } & \bar{C} X \leq \delta \\ & A X \leq b \\ & X \geq 0 \end{aligned}$$

where  $\delta$  is parameterized over the relevant part of the range of possible expected incomes i.e. from the lowest acceptable to the LP maximum.

#### 14.3.1.1.3 Formulation Choice

Markowitz's (1959) and Freund's (1956) formulations yield identical efficient frontiers; however, we favor Freund's (1956) formulation (a weighted multi-objective tradeoff model) due to a perceived incompatibility of the Markowitz formulation with model use as argued in the multi-objective chapter. Briefly, models are usually formulated for comparative statics analysis of a related series of problems. This type of analysis involves changes in the  $S$ ,  $\bar{C}$ ,  $A$  and  $b$  parameters. In such an analysis, we feel it is not desirable to give alternative efficient frontiers; rather, we feel it is desirable to give specific plans (i.e.,  $X$  variable values) for the  $S$ ,  $\bar{C}$ ,  $A$  and  $b$  settings. Using the above E-V models one would first need to select either a numerical value for  $M$  or one for  $\delta$ . A value of  $M$  so adopted is largely a function of the decision makers' preference between income and risk (see Freund (1956) or Bussey for theoretical development of this point). The value of  $\delta$  adopted will be a function of both the risk-income tradeoff and the values of  $\bar{C}$ ,  $S$ ,  $A$ , and  $b$ . Thus, the attainability of a given choice  $\delta$  would change with alterations in these parameters. On the other hand,  $M$  expresses a "pure" measure of the risk-tradeoff and is more likely to be relevant for different parameter values. Thus, we prefer the Freund (1956) formulation.

#### 14.3.1.1.4 Characteristics of E-V Model Optimal Solutions

Properties of optimal E-V solutions may be examined via the Kuhn-Tucker conditions. Given the problem

$$\begin{aligned} \text{Max } & \bar{C} X \text{ \& } M X^T S X \\ \text{s.t. } & A X \leq b \\ & X \geq 0 \end{aligned}$$

Its Lagrangian function is

$$\hat{a}(X, \mu) = \bar{C} X \text{ \& } M X^T S X \text{ \& } \mu(A X \leq b)$$

and the Kuhn-Tucker conditions are



$$\begin{array}{rcl}
\bar{M} / M X & = & \bar{C} - 2 N X' S + \mu A \quad \$ \quad 0 \\
(\bar{M} / M X) X & = & (\bar{C} - 2 N X' S + \mu A) X \quad \$ \quad 0 \\
X & & \$ \quad 0 \\
\bar{M} / M \mu & = & \mu (A X + b) \quad \$ \quad 0 \\
\mu (\bar{M} / M \mu) & = & \mu (A X + b) \quad \$ \quad 0 \\
\mu & & \$ \quad 0
\end{array}$$

where  $\mu$  is the vector of dual variables (Lagrangian multipliers) associated with the primal constraint  $AX \leq b$ .

A cursory examination of these conditions indicates two things. First, the solution permits more variables to be nonzero than would a LP basic solution. This occurs since variables can be nonzero to satisfy the  $n$  potential conditions  $\bar{M} / M X = 0$  and the  $m$  conditions where  $AX = b$  or  $\mu = 0$ . Thus, the solution can have more nonzero variables than constraints. Second, the  $\bar{M} / M X$  equation relates resource cost ( $\mu$ ) with marginal revenue ( $\bar{C}$ ) and a marginal cost of bearing risk ( $-2 M X' S$ ). Consequently, the optimal shadow prices are risk adjusted as are the optimal decision variable values.

#### 14.3.1.1.5 E-V Model Use - Theoretical Concerns

Use of the E-V model has been theoretically controversial. Expected utility theory (von Neumann and Morgenstern) provides the principal theoretical basis for choice under uncertainty. Debate has raged, virtually since the introduction of E-V analysis, on the conditions under which an E-V model makes choices equivalent to expected utility maximization. Today the general agreement is that maximizing the E-V problem is equivalent to maximizing expected utility when one of two conditions hold: 1) the underlying income distribution is normal - which requires a normal distribution of the  $c_i$  and the utility function is exponential (Freund, 1956; Bussey)<sup>10</sup>, and 2) the underlying distributions satisfy Meyer's location and scale restrictions. In addition, Tsiang (1972, 1974) has shown that E-V analysis provides an acceptable approximation of the expected utility choices when the risk taken is small relative to total initial wealth. The E-V frontier has also been argued to be appropriate under quadratic utility (Tobin). There have also been empirical studies (Levy and Markowitz; Kroll, et al.; and Reid and Tew) wherein the closeness of E-V to expected utility maximizing choices has been shown.

#### 14.3.1.1.6 Specification of the Risk Aversion Parameter

E-V models need numerical risk aversion parameters ( $M$ ). A number of approaches have been used for parameter specification. First, one may avoid specifying a value and derive the efficient frontier. This

---

<sup>10</sup> Normality probably validates a larger class of utility functions but only the exponential case has been worked out.

involves solving for many possible risk aversion parameters. Second, one may derive the efficient frontier and present it to a decision maker who picks an acceptable point (ideally, where his utility function and the E-V frontier are tangent) which in turn identifies a specific risk aversion parameter (Candler and Boeljhe). Third, one may assume that the E-V rule was used by decision makers in generating historical choices, and can fit the risk aversion parameter as equal to the difference between marginal revenue and marginal cost of resources, divided by the appropriate marginal variance (Weins). Fourth, one may estimate a risk aversion parameter such that the difference between observed behavior and the model solution is minimized (as in Brink and McCarl (1979) or Hazell et al. (1983)). Fifth, one may subjectively elicit a risk aversion parameter (see Anderson, et al. for details) and in turn fit it into the objective function (i.e., given a Pratt risk aversion coefficient and assuming exponential utility implies the E-V M equals ½ the Pratt risk aversion coefficient [Freund, 1956 or Bussey]). Sixth, one may transform a risk aversion coefficient from another study or develop one based on probabilistic assumptions (McCarl and Bessler).

The E-V model has a long history. The earliest application appears to be Freund's (1956). Later, Heady and Candler; McFarquhar; and Stovall all discussed possible uses of this methodology. A sample of applications includes those of Brainard and Cooper; Lin, et al.; and Wiens. In addition, numerous references can be found in Boisvert and McCarl; Robinson and Brake; and Barry.

#### 14.3.1.2 A Linear Approximation - MOTAD

The E-V model yields a quadratic programming problem. Such problems traditionally have been harder to solve than linear programs (although McCarl and Onal argue this is no longer true). Several LP approximations have evolved (Hazell, 1971; Thomas et al; Chen and Baker; and others as reviewed in McCarl and Tice). Only Hazell's MOTAD is discussed here due to its extensive use.

The acronym MOTAD refers to Minimization of Total Absolute Deviations. In the MOTAD model, absolute deviation is the risk measure. Thus, the MOTAD model depicts tradeoffs between expected income and the absolute deviation of income. Minimization of absolute values is discussed in the nonlinear approximations chapter. Briefly reviewing, absolute value may be minimized by constraining the terms whose absolute value is to be minimized ( $D_k$ ) equal to the difference of two non-negative variables ( $D_k = d_k^+ - d_k^-$ ) and in turn minimizing the sum of the new variables  $d_k^+ + d_k^-$ . Hazell(1971) used this formulation in developing the MOTAD model.<sup>11</sup>

Formally, the total absolute deviation of income from mean income under the  $k^{\text{th}}$  state of nature ( $D_k$ ) is

$$D_k = \left| \sum_j c_{kj} X_j \right| = \left( \sum_j \bar{c}_j X_j \right)^+ - \left( \sum_j \bar{c}_j X_j \right)^-$$

---

<sup>11</sup> The approach was suggested in Markowitz (1959, p. 187).

where  $c_{kj}$  is the per unit net return to  $X_j$  under the  $k^{\text{th}}$  state of nature and  $\bar{c}_j$  is the mean.

Since both terms involve  $X_j$  and sum over the same index, this can be rewritten as

$$D_k = \sum_j (c_{kj} - \bar{c}_j) X_j$$

Total absolute deviation (TAD) is the sum of  $D_k$  across the states of nature. Now introducing deviation variables to depict positive and negative deviations we get

$$\text{TAD} = \sum_k D_k = \sum_k (d_k^+ + d_k^-)$$

$$\text{where } \sum_j (c_{kj} - \bar{c}_j) X_j = d_k^+ - d_k^- \quad \text{for all } k$$

Then adding the sum of the deviation variables to the objective function the MOTAD model maximizes expected net returns less a risk aversion coefficient ( $Q$ ) times the measure of absolute deviation.

The final MOTAD formulation is

$$\begin{aligned} \text{Max} \quad & \sum_j \bar{c}_j X_j - Q \sum_k (d_k^+ + d_k^-) \\ \text{s.t.} \quad & \sum_j (c_{kj} - \bar{c}_j) X_j = d_k^+ - d_k^- \quad \text{for all } k \\ & \sum_j a_{ij} X_j \leq b_i \quad \text{for all } i \\ & X_j, d_k^+, d_k^- \geq 0 \quad \text{for all } j, k \end{aligned}$$

where  $d_k^+$  is the positive deviation of the  $k^{\text{th}}$  income occurrence from mean income and  $d_k^-$  is the associated negative deviation.<sup>12</sup>

There have been a number of additional developments regarding the MOTAD formulation. Hazell formulated a model considering only negative deviations from the mean, ignoring positive deviations. This formulation is:

$$\begin{aligned} \text{Max} \quad & \sum_j \bar{c}_j X_j - 2 \sum_k d_k^- \\ \text{s.t.} \quad & \sum_j (c_{kj} - \bar{c}_j) X_j = -d_k^- \quad \text{for all } k \\ & \sum_j a_{ij} X_j \leq b_i \quad \text{for all } i \\ & X_j, d_k^- \geq 0 \quad \text{for all } j, k \end{aligned}$$

<sup>12</sup> Note this formulation approach can be used within an E-V framework if one squares  $d^+$  and  $d^-$  in the objective function.



error, but later becomes quite close. The E-V and MOTAD frontiers correspond closely (see Figure 14.2). However, this is not adequate proof that the solutions will always be close (see Thomson and Hazell for a comparison between the methods).

#### 14.3.1.2.2 Comments on MOTAD

Many of the E-V model comments are appropriate here and will not be repeated. However, a number of other comments are in order. First, a cursory examination of the MOTAD model might lead one to conclude covariance is ignored. This is not so. The deviation equations add across all the variables, allowing negative deviation in one variable to cancel positive deviation in another. Thus, in minimizing total absolute deviation the model has an incentive to "diversify", taking into account covariance.

Second, the equivalence of the total negative and total absolute deviation formulations depends critically upon deviation symmetry. Symmetry will occur whenever the deviations are taken from the mean. This, however, implies that the mean is the value expected for each observation. This may not always be the case. When the value expected is not the mean, then moving averages or other expectation models should be used instead of the mean (see Brink and McCarl, or Young). In such cases, the deviations are generally non-symmetric and consideration must be given to an appropriate measure of risk. For example, Brink and McCarl use a mean negative deviation formulation with a moving average expectation.

Third, most MOTAD applications use approximated standard errors as a measure of risk. When using such a measure, the risk aversion parameters can be interpreted as the number of standard errors one wishes to discount income. Coupling this with a normality assumption permits one to associate a confidence limit with the risk aversion parameter. For example, a risk aversion parameter equal to one means that level of income which occurs at one standard error below the mean is maximized. Assuming normality, this level of income is 84% sure to occur.

Fourth, one must have empirical values for the risk aversion parameter. All the E-V approaches are applicable to its discovery. The most common approach with MOTAD models has been based on observed behavior. The procedure has been to: a) take a vector of observed solution variables, (i.e. acreages); b) parameterize the risk aversion coefficient in small steps (e.g., 0.25) from 0 to 2.5, at each point computing a measure of the difference between the model solution and observed behavior; and c) select the risk aversion parameter value for which the smallest dispersion is found between the model solution values and the observed values (for examples see Hazell et al.; Brink and McCarl; Simmons and Pomareda; or Nieuwoudt, et al.).

Fifth, the MOTAD model does not have a general direct relationship to a theoretical utility function. Some authors have discovered special cases under which there is a link (see Johnson and Boeljhe(1981,1983) and their subsequent exchange with Buccola). Largely, the MOTAD model has been presented as an approximation to the E-V model. However, with the advances in nonlinear programming

algorithms the approximation motivation is largely gone (McCarl and Onal), but MOTAD may have application to non-normal cases (Thomson and Hazell).

Sixth, McCarl and Bessler derive a link between the E-standard error and E-V risk aversion parameters as follows:

Consider the models

$$\begin{array}{l} \text{Max } cX + Q F^2(X) \\ \text{s.t. } AX \leq b \\ X \geq 0 \end{array} \quad \text{versus} \quad \begin{array}{l} \text{Max } cX + \gamma F(X) \\ \text{s.t. } AX \leq b \\ X \geq 0 \end{array}$$

The first order conditions assuming X is nonzero are

$$c + 2Q F(X) \frac{dF(X)}{dX} - \lambda A = 0 \quad \text{and} \quad c + \gamma \frac{dF(X)}{dX} - \lambda A = 0$$

For these two solutions to be identical in terms of X and λ, then

$$\lambda = \frac{\gamma}{2 F(X)}$$

Thus, the E-V risk aversion coefficient will equal the E-standard error model risk aversion coefficient divided by twice the standard error. This explains why E-V risk aversion coefficients are usually very small (i.e., an E-standard error risk aversion coefficient usually ranges from 0 - 3 which implies when the standard error of income is \$10,000 the E-V risk aversion coefficient range of 0 - .000015). Unfortunately, since γ is a function of F which depends on X, this condition must hold ex post and cannot be imposed a priori. However, one can develop an approximate a priori relationship between the risk aversion parameters given an estimate of the standard error.

The seventh and final comment regards model sensitivity. Schurle and Erven show that several plans with very different solutions can be feasible and close to the plans on the efficient frontier. Both results place doubt on strict adherence to the efficient frontier as a norm for decision making. (Actually the issue of near optimal solutions is much broader than just its role in risk models.) The MOTAD model has been rather widely used. Early uses were by Hazell (1971); Hazell and Scandizzo; Hazell et al. (1983); Simmons and Pomareda; and Nieuwoudt, et al. In the late 1970's the model saw much use. Articles from 1979 through the mid 1980s in just the American Journal of Agricultural Economics include Gebremeskel and Shumway; Schurle and Erven; Pomareda and Samayoa; Mapp, et al.; Apland, et al. (1980); and Jabara and Thompson. Boisvert and McCarl provide a recent review.

#### 14.3.1.3 Toward A Unified Model

The E-V and MOTAD models evolved before many software developments. As a consequence, the models were formulated to be easily solved with 1960's and 70's software. A more extensive unified model formulation is possible today. The E-standard error form of this model is as follows

$$\begin{array}{ll}
\text{Max} & \overline{\text{Inc}} \ \& \ M \left\{ \sum_k p_k [(d_k^{\%})^2 + (d_k^{\&})^2] \right\}^{0.5} \\
\text{s.t.} & \sum_j a_{ij} X_j \leq b_i \quad \text{for all } i \\
& \sum_j c_{kj} X_j \leq \text{Inc}_k \quad \text{for all } k \\
& \sum_k p_k \text{Inc}_k \leq \overline{\text{Inc}} \\
& \text{Inc}_k \leq \overline{\text{Inc}} + d_k^{\%} \quad \text{for all } k \\
& \text{Inc}_k \leq \overline{\text{Inc}} - d_k^{\&} \quad \text{for all } k \\
& \text{Inc}_k \leq \overline{\text{Inc}} \leq \text{Inc}_k + d_k^{\%} \quad \text{for all } k \\
& \text{Inc}_k \leq \overline{\text{Inc}} \leq \text{Inc}_k - d_k^{\&} \quad \text{for all } k
\end{array}$$

In this model the resource constraints continue to appear. But we introduce a new variable ( $\text{Inc}_k$ ) which is income under state of nature  $k$ . This is equated with income arising under the  $k^{\text{th}}$  state of nature. In turn, a variable is entered for average income ( $\overline{\text{Inc}}$ ) which is equated to the probabilities ( $p_k$ ) times the income levels. This variable appears in the objective function reflecting expected income maximization. Finally, deviations between the average and state of nature dependent income levels are treated in deviation constraints where  $d_k^+$  indicates income above the average level whereas  $d_k^-$  indicates shortfalls. The objective function is then modified to include the probabilities and deviation variables. Several possible objective function formulations are possible. The objective function formulation above is E-standard error without approximation. Note that the term in parentheses contains the summed, probabilistically weighted, squared deviations from the mean and is by definition equal to the variance. In turn, the square root of this term is the standard deviation and  $M$  would be a risk aversion parameter which would range between zero and 2.5 in most circumstances (as explained in the MOTAD section).

This objective function can also be reformulated to be equivalent to either the MOTAD or E-V cases. Namely, in the E-V case if we drop the 0.5 exponent then the bracketed term is variance and the model would be E-V. Similarly, if we drop the 0.5 exponent and do not square the deviation variables then a MOTAD model arises.

This unifying framework shows how the various models are related and indicates that covariance is considered in any of the models. An example is not presented here although the files UNIFY, EV2 and MOTAD2 give GAMS implementations of the unified E-standard error, E-V and MOTAD versions. The resultant solutions are identical to the solution for E-V and MOTAD examples and are thus not discussed further.

#### 14.3.1.4 Safety First

Roy posed a different approach to handling objective function uncertainty. This approach, the Safety First model, assumes that decision makers will choose plans to first assure a given safety level for income. The formulation arises as follows: assume the model income level under all  $k$  states of nature ( $\sum c_{kj} X_j$ )

must exceed the safety level (S). This can be assured by entering the constraints

$$c_{kj} X_j \geq S \quad \text{for all } k$$

The overall problem then becomes

$$\begin{aligned} \text{Max} \quad & \sum_j \bar{c}_j X_j \\ \text{s.t.} \quad & \sum_j a_{ij} X_j \leq b_i \quad \text{for all } i \\ & \sum_j c_{kj} X_j \geq S \quad \text{for all } k \\ & X_j \geq 0 \quad \text{for all } j \end{aligned}$$

where S is the safety level.

#### 14.3.1.4.1 Example

A formulation using the data from the E-V example and a safety level of S is given in Table 14.8 and a GAMS implementation is in the file SAFETY. This example was solved for safety levels ranging from -\$100 to +\$50. The solution (Table 14.9) at S = \$100 gives the profit maximizing linear programming solution. As the safety level is increased the solutions reflect a diversification between X<sub>3</sub> and X<sub>2</sub>. These solutions exhibit the same sort of behavior as in the previous examples. As the safety level increases a more diversified solution arises with an accompanying reduction in risk and a decrease in expected value. For example at S = \$50 the mean has dropped from \$148.00 to \$135.00, but the standard error is cut by more than two-thirds.

#### 14.3.1.4.2 Comments

The safety first model has not been extensively used empirically although Target MOTAD as discussed in the next section is a more frequently used extension. However, the Safety First model is popular as an analytical model in characterizing decision making. For a review and more extensive discussion see Barry.

#### 14.3.1.5 Target MOTAD

The Target MOTAD formulation developed by Tauer, incorporates a safety level of income while also allowing negative deviations from that safety level. Given a target level of T, the formulation is



$$\begin{aligned}
\text{Max} \quad & \sum_j \bar{c}_j X_j \\
\text{s.t.} \quad & \sum_j a_{ij} X_j \leq b_i \quad \text{for all } i \\
& \sum_j c_{kj} X_j + \text{Dev}_k \leq T \quad \text{for all } k \\
& \sum_k p_k \text{Dev}_k \leq \delta \\
& X_j, \text{Dev}_k \geq 0 \quad \text{for all } j, k
\end{aligned}$$

All definitions are as above except  $p_k$  is the probability of the  $k^{\text{th}}$  state of nature;  $T$  is the target income level (somewhat analogous to  $S$  in the safety first model); the variable  $\text{Dev}_k$  is the negative deviation of income, allowing income under the  $k^{\text{th}}$  state of nature to fall below target income; and  $\delta$  is the maximum average income shortfall permitted. The equation containing  $T$  gives the relationship between income under the  $k^{\text{th}}$  state of nature and a target income level. The variable  $\text{Dev}_k$  is non-zero if the  $k^{\text{th}}$  income result falls below  $T$ . The constraint with the right hand side of  $\delta$  limits the average shortfall. Thus, the Target MOTAD model has two parameters relating to risk ( $T$  and  $\delta$ ) which must be specified. These, in turn, can be parameterized to yield different risk solutions.

#### 14.3.1.5.1 Example

Using the data from the earlier examples and assuming each state of nature is equally probable ( $P_k = 1/10$ ) yields the formulation given in Table 14.10 and the GAMS formulation is in the file TARGET.

The Target MOTAD example was solved with a safety level of \$120.00 with the allowable deviation from the safety level varied from allowing as much as \$120.00 average deviation to as little as \$3.60. The solution behavior (Table 14.11) again largely mirrors that observed in the prior examples. Namely, when a large deviation is allowed, the profit maximizing solution is found, but as the allowable deviation gets smaller, then  $X_2$  enters and then finally  $X_1$ . Again a sacrifice in expected income yields less risk.

#### 14.3.1.5.2 Comments

Target MOTAD has not been applied as widely as other risk programming models. However, it is consistent with second degree stochastic dominance (Tauer). Use of Target MOTAD requires specification of two parameters,  $T$  and  $\delta$ . No attempt has been made to determine consistency between a  $T, \delta$  choice and the Arrow-Pratt measure of risk aversion. Nor is there theory on how to specify  $T$  and  $\delta$ . The target MOTAD and original MOTAD models can be related. If one makes  $\delta$  a variable with a cost in the objective function and makes the target level a variable equal to expected income, this becomes the MOTAD model.

Another thing worth noting is that the set of Target MOTAD solutions are continuous so that there is an infinite number of solutions. In the example, any target deviation between \$24.00 and \$12.00 would be

a unique solution and would be a convex combination of the two tabled solutions.

McCamley and Kliebenstein outline a strategy for generating all target MOTAD solutions, but it is still impossible to relate these solutions to more conventional measures of risk preferences.

Target MOTAD has been used in a number of contexts. Zimet and Spreen formulate a farm production implementation while Curtis et al., and Frank et al., studied crop marketing problems.

#### 14.3.1.6 DEMP

Lambert and McCarl (1985) introduced the Direct Expected Maximizing Nonlinear Programming (DEMP) formulation, which maximizes the expected utility of wealth. DEMP was designed as an alternative to E-V analysis, relaxing some of the restrictions regarding the underlying utility function. The basic DEMP formulation requires specification of a utility of wealth function  $U(W)$ , a level of initial wealth ( $W_0$ ), and the probability distribution of the objective function parameters ( $C_{kj}$ ). The basic formulation is

$$\begin{aligned} \text{Max } & \sum_k p_k U(W_k) \\ \text{s.t. } & \sum_j a_{ij} X_j \leq b_i \quad \text{for all } i \\ & W_k \leq \sum_j c_{kj} X_j + W_0 \quad \text{for all } k \\ & W_k \geq 0 \quad \text{for all } k \\ & X_j \geq 0 \quad \text{for all } j \end{aligned}$$

where  $p_k$  is the probability of the  $k^{\text{th}}$  state of nature;

$W_0$  is initial wealth;

$W_k$  is the wealth under the  $k^{\text{th}}$  state of nature; and

$c_{kj}$  is the return to one unit of the  $j^{\text{th}}$  activity under the  $k^{\text{th}}$  state of nature.

##### 14.3.1.6.1 Example

Suppose an individual has the utility function for wealth of the form  $U = (W)^{\text{power}}$  with an initial wealth ( $W_0$ ) of 100, and is confronted with the decision problem data as used in the E-V example. The relevant DEMP formulation appears in Table 14.12 with the solution for varying values of the exponent appearing in Table 14.13. The GAMS formulation is called DEMP.

The example model was solved for different values of the exponent (power). The exponent was varied from 0.3 to 0.0001. As this was varied, the solution again transitioned out of sole reliance on stock three into reliance on stocks two and three. During the model calculations, transformations were done on the shadow price to convert it into dollars. Following Lambert and McCarl, this may be converted into an approximate value in dollar space by dividing by the marginal utility of average income i.e., dividing the

shadow prices by the factor.

$$\mu^k = \mu / \frac{M U(\bar{W})}{M W}$$

Preckel, Featherstone, and Baker discuss a variant of this procedure.

#### 14.3.1.6.2 Comments

The DEMP model has two important parts. First, note that the constraints involving wealth can be rearranged to yield

$$W_k = W_0 + \sum_j c_{kj} X_j$$

This sets wealth under the  $k^{\text{th}}$  state of nature equal to initial wealth plus the increment to wealth due to the choice of the decision variables.

Second, note that the objective function equals expected utility. Thus the formulation maximizes expected utility using the empirical distribution of risk without any distributional form assumptions and an explicit, exact specification of the utility function.

Kaylen, et al., employ a variation of DEMP where the probability distributions are of a known continuous form and numerical integration is used in the solution. The DEMP model has been used by Lambert and McCarl(1989); Lambert; and Featherstone et al.

Yassour, et al., present a related expected utility maximizing model called EUMGF, which embodies both an exponential utility function and distributional assumptions. They recognize that the maximization of expected utility under an exponential utility function is equivalent to maximization of the moment generating function (Hogg and Craig) for a particular probability distribution assumption. Moment generating functions have been developed analytically for a number of distributions, including the Binomial, Chi Square, Gamma, Normal and Poisson distributions. Collender and Zilberman and Moffit et al. have applied the EUMGF model. Collender and Chalfant have proposed a version of the model no longer requiring that the form of the probability distribution be known.

#### 14.3.1.7 Other Formulations

The formulations mentioned above are the principal objective function risk formulations which have been used in applied mathematical programming risk research. However, a number of other formulations have been proposed. Alternative portfolio models such as those by Sharpe; Chen and Baker; Thomas et al.(1972) exist. Other concepts of target income have also been pursued (Boussard and Petit) as have models based upon game theory concepts (McInerney [1967, 1969]; Hazell and How; Kawaguchi and Maruyama; Hazell(1970); Agrawal and Heady; Maruyama; and Low) and Gini coefficients (Yitzhaki). These have all experienced very limited use and are therefore not covered herein.

### **14.3.2 Right Hand Side Risk**

Risk may also occur within the right hand side (RHS) parameters. The most often used approach to RHS risk in a nonrecourse setting is chance-constrained programming. However, Paris(1979) has tried to introduce an alternative.

#### 14.3.2.1 Chance Constrained Programming

The chance-constrained formulation was introduced by Charnes and Cooper and deals with uncertain RHS's assuming the decision maker is willing to make a probabilistic statement about the frequency with which constraints need to be satisfied. Namely, the probability of a constraint being satisfied is greater than or equal to a prespecified value  $\alpha$ .

$$P \left( \sum_j a_{ij} X_j \leq b_i \right) \geq \alpha$$

If the average value of the RHS ( $\bar{b}_i$ ) is subtracted from both sides of the inequality and in turn both sides are divided by the standard deviation of the RHS ( $F_{b_i}$ ) then the constraint becomes

$$P \left[ \frac{\sum_j a_{ij} X_j - \bar{b}_i}{F_{b_i}} \leq \frac{(b_i - \bar{b}_i)}{F_{b_i}} \right] \geq \alpha$$

Those familiar with probability theory will note that the term

$$\frac{(b_i - \bar{b}_i)}{F_{b_i}}$$

gives the number of standard errors that  $b_i$  is away from the mean. Let  $Z$  denote this term.

When a particular probability limit ( $\alpha$ ) is used, then the appropriate value of  $Z$  is  $Z_\alpha$  and the constraint becomes

$$P \left[ \frac{\sum_j a_{ij} X_j - \bar{b}_i}{F_{b_i}} \leq Z_\alpha \right] \geq \alpha$$

Assuming we discount for risk, then the constraint can be restated as

$$\sum_j a_{ij} X_j \leq \bar{b}_i + Z_\alpha F_{b_i}$$

which states that resource use ( $\sum_j a_{ij} X_j$ ) must be less than or equal to average resource availability less the standard deviation times a critical value which arises from the probability level.

Values of  $Z_\alpha$  may be determined in two ways: a) by making assumptions about the form of the probability distribution of  $b_i$  (for example, assuming normality and using values for the lower tail from a standard normal probability table); or b) by relying on the conservative estimates generated by using Chebyshev's inequality, which states the probability of an estimate falling greater than  $M$  standard deviations away from the mean is less than or equal to one divided by  $M^2$ . Using the Chebyshev inequality

one needs to solve for that value of  $M$  such that  $(1-\alpha)$  equals  $1/M^2$ . Thus, given a probability  $\alpha$ , the Chebyshev value of  $Z$  is given by the equation  $Z=(1-\alpha)^{-0.5}$ . Following these approaches, if one wished an 87.5 percent probability, a normality assumption would discount 1.14 standard deviations and an application of the Chebyshev inequality would lead to a discount of 2.83 standard deviations. However, one should note that the Chebyshev bound is often too large.

#### 14.3.2.1.1 Example

The example problem adopted for this analysis is in the context of the resource allocation problem from Chapter V. Here three of the four right hand sides in that problem are presumed to be stochastic with the distribution as given in Table 14.14. Treating each of these right hand side observations as equally likely, the mean value equals those numbers that were used in the resource allocation problem and their standard errors respectively are as given in Table 14.14. Then the resultant chance constrained formulation is

$$\begin{array}{ll}
 \text{Max} & 67X_1 + 66X_2 + 66.3X_3 + 80X_4 + 78.5X_5 + 78.4X_6 \\
 \text{s.t} & 0.8X_1 + 1.3X_2 + 0.2X_3 + 1.2X_4 + 1.7X_5 + 0.5X_6 \# 140 - 9.63 Z \\
 & 0.5X_1 + 0.2X_2 + 1.3X_3 + 0.7X_4 + 0.3X_5 + 1.5X_6 \# 90 - 3.69 Z \\
 & 0.4X_1 + 0.4X_2 + 0.4X_3 + X_4 + X_5 + X_6 \# 120 - 8.00 Z \\
 & X_1 + 1.05X_2 + 1.1X_3 + 0.8X_4 + 0.82X_5 + 0.84X_6 \# 125
 \end{array}$$

The GAMS implementation is the file CHANCE. The solutions to this model were run for  $Z$  values corresponding to 0, 90, 95, and 99 percent confidence intervals under a normality assumption. The right hand sides and resultant solutions are tabled in Table 14.15. Notice as the  $Z$  value is increased, then the value of the uncertain right hand side decreases. In turn, production decreases as does profit. The chance constrained model discounts the resources available, so one is more certain that the constraint will be met. The formulation also shows how to handle simultaneous constraints. Namely the constraints may be treated individually. Note however this requires an assumption that the right hand sides are completely dependent. The results also show that there is a chance of the constraints being exceeded but no adjustment is made for what happens under that circumstance.

#### 14.3.2.1.2 Comments

Despite the fact that chance constrained programming (CCP) is a well known technique and has been applied to agriculture (e.g., Boisvert, 1976; Boisvert and Jensen, 1973; and Danok et al., 1980) and water management (e.g., Eisel; Loucks; and Maji and Heady) its use has been limited and controversial. See the dialogue in Blau; Hogan, et al.; and Charnes and Cooper (1959).

The major advantage of CCP is its simplicity; it leads to an equivalent programming problem of about the same size and the only additional data requirements are the standard errors of the right hand side. However, its only decision theoretic underpinning is Simon's principle of satisficing (Pfaffenberger and

Walker).

This CCP formulation applies when either one element of the right hand side vector is random or when the distribution of multiple elements is assumed to be perfectly correlated. The procedure has been generalized to other forms of jointly distributed RHS's by Wagner (1975). A fundamental problem with chance constrained programming (CCP) is that it does not indicate what to do if the recommended solution is not feasible. From this perspective, Hogan et al., (1981), conclude that "... there is little evidence that CCP is used with the care that is necessary" (p. 698) and assert that recourse formulations should be used.

#### 14.3.2.2 A Quadratic Programming Approach

Paris(1979) proposed a quadratic programming model which permits RHS risk in an E-V context. In contrast to chance constrained programming, the formulation treats inter-dependencies between the RHS's. The formulation is developed through an application of non-linear duality theory and is

$$\begin{aligned} \text{Max } & \bar{c}X + N X^T S_c X + 1 Y^T S_b Y \\ \text{s.t. } & AX + 1 S_b Y \leq \bar{b} \\ & X \geq 0, \quad Y \geq 0 \end{aligned}$$

where  $X$  is the vector of activities;  $N$  and  $1$  are risk aversion coefficients with respect to variance in returns and the RHS.  $S_c$  and  $S_b$  are variance-covariance matrices of returns and the RHS's, respectively;  $Y$  is the vector of dual variables,  $A$  is the matrix of technical coefficients, and  $\bar{b}$  is the vector of expected values of the RHS's.

This primal model explicitly contains the dual variables and the variance-covariance matrix of the RHS's. However, the solutions are not what one might expect. Namely, in our experience, as right hand side risk aversion increases, so does expected income. The reason lies in the duality implications of the formulation. Risk aversion affects the dual problem by making its objective function worse. Since the dual objective function value is always greater than the primal, a worsening of the dual objective via risk aversion improves the primal. A manifestation of this appears in the way the risk terms enter the constraints. Note given positive  $1$  and  $S_b$ , then the sum involving  $1$  and  $Y$  on the left hand side augments the availability of the resources. Thus, under any nonzero selection of the dual variables, as the risk aversion parameter increases so does the implicit supplies of resources. Dubman et al., and Paris(1989) debate these issues, but the basic flaw in the formulation is not fixed. Thus we do not recommend use of this formulation and do not include an example.

#### **14.3.3 Technical Coefficient Risk**

Risk can also appear within the matrix of technical coefficients. Resolution of technical coefficient uncertainty in a non-recourse setting has been investigated through two approaches. These involve an E-V like procedure (Merrill), and one similar to MOTAD (Wicks and Guise).

##### 14.3.3.1 Merrill's Approach

Merrill formulated a nonlinear programming problem including the mean and variance of the risky  $a_{ij}$ 's into the constraint matrix. Namely, one may write the mean of the risky part as  $\sum_j \bar{a}_{ij} X_j$  and its variance as  $\sum_j \sum_n X_j X_n F_{ijn}$  where  $\bar{a}_{ij}$  is the mean value of the  $a_{ij}$ 's and  $F_{ijn}$  is the covariance of the  $a_{ij}$  coefficients for activities  $n$  and  $j$  in row  $i$ . Thus, a constraint containing uncertain coefficients can be rewritten as

$$\sum_j \bar{a}_{ij} X_j + N \sum_{jn} X_j X_n F_{ijn} \leq b_i \quad \text{for all } i$$

or, using standard deviation,

$$\sum_j \bar{a}_{ij} X_j + N \left( \sum_{jk} X_j X_k F_{ijn} \right)^{0.5} \leq b_i \quad \text{for all } i$$

Note that the term involving  $F_{ijn}$  is added inflating resource use above the average to reflect variability, thus a safety cushion is introduced between average resource use and the reserve limit. The parameter  $N$  determines the amount of safety cushion to be specified exogenously and could be done using distributional assumptions (such as normality) or Chebyshev's inequality as argued in McCarl and Bessler. The problem in this form requires usage of nonlinear programming techniques.

Merrill's approach has been unused largely since it was developed at a time when it was incompatible with available software. However, the MINOS algorithm in GAMS provides capabilities for handling the nonlinear constraint terms (although solution times may be long -- McCarl and Onal). Nevertheless the simpler Wicks and Guise approach discussed below is more likely to be used. Thus no example is given.

#### 14.3.3.2 Wicks and Guise Approach

Wicks and Guise provided a LP version of an uncertain  $a_{ij}$  formulation based on Hazell(1971) and Merrill's models. Specifically, given that the  $i^{\text{th}}$  constraint contains uncertain  $a_{ij}$ 's, the following constraints may be set up.

$$\begin{aligned} \sum_j \bar{a}_{ij} X_j + N D_i &\leq b_i \\ \sum_j (a_{kij} - \bar{a}_{ij}) X_j &\leq d_{ki}^{\%} \\ \sum_j (d_{ki}^{\%} - a_{kij}) X_j &\leq d_{ki}^{\&} \\ \sum_k (d_{ki}^{\%} + d_{ki}^{\&}) &\leq D_i \end{aligned} \quad \text{for all } k$$

Here the first equation relates the mean value of uncertain resource usage plus a risk term ( $N D_i$ ) to the right hand side, while the second computes the deviation ( $a_{kij} - \bar{a}_{ij}$ ) incurred from the  $k^{\text{th}}$  joint observation on all  $a_{ij}$ 's and sums it into a pair of deviation variables ( $d_{ki}^{\%}$ ,  $d_{ki}^{\&}$ ). These deviation variables are in turn summed into a measure of total absolute deviation ( $D_i$ ) by the third equation. The term  $N D_i$  then gives the risk adjustment to the mean resource use in constraint  $i$  where  $N$  is a coefficient of risk aversion.

The Wicks and Guise formulation is essentially this; however, Wicks and Guise convert the total absolute deviation into an estimate of standard deviation using a variant of the Fisher constant but we will use the one discussed above

$$D - F = 0$$

where  $\delta = (B/(2n(n-1)))^{.5}$  and  $F$  is the standard error approximation. The general Wicks Guise formulation is

$$\begin{aligned} \text{Max} \quad & \sum_j c_j X_j \\ \text{s.t.} \quad & \sum_j \bar{a}_{ij} X_j \leq N F_i + b_i \quad \text{for all } i \\ & \sum_j (a_{kij} \delta + \bar{a}_{ij}) X_j \leq d_{ki}^{\%} + d_{ki}^{\&} \quad \text{for all } i,k \\ & \sum_k (d_{ki}^{\%} + d_{ki}^{\&}) \leq D_i \leq 0 \quad \text{for all } i \\ & \sum_j X_j, \quad d_{ki}^{\%}, \quad d_{ki}^{\&}, \quad D_i, \quad F_i \geq 0 \quad \text{for all } j,k,i \end{aligned}$$

#### 14.3.3.2.1 Example

Suppose we introduce ingredient uncertainty in the context of the feed problem as discussed in Chapter V. Suppose one is using three feed ingredients corn, soybeans, and wheat while having to meet energy and protein requirements. However, suppose that there are four states of nature for energy and protein nutrient content as given in Table 14.16. Assume that the unit price of corn is 3 cents, soybeans 6 cents, and wheat 4 cents and that the energy requirements are 80% of the unit weight of the feed while the protein requirement is 50%. In turn, the GAMS formulation of this is called WICKGUIS and a tableau is given in Table 14.17.

The solution to the Wicks Guise example model are given in Table 14.18. Notice in this table when the risk aversion parameter is 0 then the model feeds corn and wheat, but as the risk aversion parameter increases the model first reduces its reliance on corn and increases wheat, but as the risk aversion parameter gets larger and larger one begins to see soybeans come into the answer. Notice across these solutions, risk aversion generally increases the average amount of protein with reductions in protein variability. As the risk aversion parameter increases, the probability of meeting the constraint increases. Also notice that the shadow price on protein monotonically increases indicating that it is the risky ingredient driving the model adjustments. Meanwhile average energy decreases, as does energy variation and the shadow price on energy is zero, indicating there is sufficient energy in all solutions.

#### 14.3.3.2.2 Comments

The reader should note that the deviation variables do not work well unless the constraint including



the risk adjustment is binding. However, if it is not binding, then the uncertainty does not matter.

The Wicks and Guise formulation has not been widely used. Other than the initial application by Wicks and Guise the only other application we know of is that of Tice.

Several other efforts have been made regarding  $a_{ij}$  uncertainty. The method used in Townsley and later by Chen (1973) involves bringing a single uncertain constraint into the objective function. The method used in Rahman and Bender involves developing an over-estimate of variance.

#### **14.3.4 Multiple Sources of Risk**

Many problems have C's, A's and b's which are simultaneously uncertain. The formulations above may be combined to handle such a case. Thus, one could have a E-V model with several constraints handled via the Wicks Guise and/or chance constrained techniques. There are also techniques for handling multiple sources of risk under the stochastic programming with recourse topic.

#### **14.4 Sequential Risk-Stochastic Programming with Recourse**

Sequential risk arises as part of the risk as time goes on and adaptive decisions are made. Consider the way that weather and field working time risks are resolved in crop farming. Early on, planting and harvesting weather are uncertain. After the planting season, the planting decisions have been made and the planting weather has become known, but harvesting weather is still uncertain. Under such circumstances a decision maker would adjust to conform to the planting pattern but would still need to make harvesting decisions in the face of harvest time uncertainty. Thus sequential risk models must depict adaptive decisions along with fixity of earlier decisions (a decision maker cannot always undo earlier decisions such as planted acreage). Nonsequential risk, on the other hand, implies that a decision maker chooses a decision now and finds out about all sources of risk later.

All the models above are nonsequential risk models. Stochastic programming with recourse (SPR) models are used to depict sequential risk. The first of the models was originally developed as the "two-stage" LP formulation by Dantzig (1955). Later, Cocks devised a model with N stages, calling it discrete stochastic programming. Over time, the whole area has been called stochastic programming with recourse (SPR). We adopt this name.

##### **14.4.1 Two stage SPR formulation**

Suppose we set up a two stage SPR formulation. Such formulations contain a probability tree (Figure 14.3). The nodes of the tree represent decision points. The branches of the tree represent alternative possible states. A two stage model has one node and set of decision variables ( $X$ ) at the first stage, with the second stage containing branches associated with the resolved uncertainty from the first stage and associated decision nodes ( $Z_k$ ).

Suppose the variables  $X_j$  indicate the amount of the  $j^{\text{th}}$  alternative which is employed in the first stage. There is an associated set of resource constraints where the per unit usage of the  $i^{\text{th}}$  resource by  $X_j$  is  $a_{ij}$  and

the endowment of the resources  $b_i$ . Suppose that the outcome of  $X_j$  is uncertain and dependent on state of nature where the quantity of the  $m^{\text{th}}$  output item produced is  $d_{mjk}$  where  $k$  designates state of nature. Let us also define  $c_j$  as the objective function coefficient for  $X_j$ . In the second stage, the variables are  $Z_{nk}$ , where  $n$  represents the  $n^{\text{th}}$  alternative for production and  $k$  identifies state of nature. Here we have different decision variables for each second stage state of nature. For example, we have the amount of stock sold if the market has been moving up and the amount of stock sold if the market is moving down, with second stage decisions that depend upon the resultant state of nature after the first stage. We also have parameters which give the amount of the  $m^{\text{th}}$  output item carrying over from stage one ( $f_{mnk}$ ) while  $g_{wnk}$  gives the amount of the  $w^{\text{th}}$  resource utilized by  $Z_{nk}$ . Finally, the objective function parameter for  $Z_{nk}$  is  $e_{nk}$ . The model also requires definition of right hand side parameters where  $s_{wk}$  is the amount of the  $w^{\text{th}}$  resource available under the  $k^{\text{th}}$  state of nature. In setting this model up we also define a set of accounting variables  $Y_k$ , which add up income under the states of nature. Finally suppose  $p_k$  gives the probability of state  $k$ . The composite model formulation is

$$\begin{aligned}
 \text{Max} \quad & \sum_k p_k Y_k \\
 \text{s.t.} \quad & Y_k \leq \sum_j c_j X_j + \sum_n e_{nk} Z_{nk} \quad \text{for all } k \\
 & \sum_j a_{ij} X_j \leq b_i \quad \text{for all } i \\
 & \sum_j d_{mjk} X_j + \sum_n f_{mnk} Z_{nk} \leq 0 \quad \text{for all } m,k \\
 & \sum_n g_{wnk} Z_{nk} \leq s_{wk} \quad \text{for all } w,k \\
 & Y_k \geq 0 \quad \text{for all } k \\
 & X_j, Z_{nk} \geq 0 \quad \text{for all } j,n,k
 \end{aligned}$$

In this problem we have income variables for each of the  $k$  states of nature ( $Y_k$ ) which are unrestricted in sign. Given that  $p_k$  is the probability of the  $k^{\text{th}}$  state of nature, then the model maximizes expected income. Note the income variable under the  $k^{\text{th}}$  state of nature is equated to the sum of the nonstochastic income from the first stage variables plus the second stage state of nature dependent profit contribution. Also note that since  $Z$  has taken on the subscript  $k$ , the decision variable value will in general vary by state of nature.

Several points should be noted about this formulation. First, let us note what is risky. In the second stage the resource endowment ( $S_{wk}$ ), constraint coefficients ( $d_{mjk}$ ,  $f_{mnk}$ ,  $g_{wnk}$ ) and objective function parameters ( $e_{nk}$ ) are dependent upon the state. Thus, all types of coefficients (RHS, OBJ and  $A_{ij}$ ) are potentially risky and their values depend upon the path through the decision tree.

Second, this model reflects a different uncertainty assumption for  $X$  and  $Z$ . Note  $Z$  is chosen with

knowledge of the stochastic outcomes; however,  $X$  is chosen a priori, with its value fixed regardless of the stochastic outcomes. Also notice that the first, third, and fourth constraints involve uncertain parameters and are repeated for each of the states of nature. This problem then has a single  $X$  solution and a  $Z$  solution for each state of nature. Thus, adaptive decision making is modeled as the  $Z$  variables are set conditional on the state of nature. Note that irreversibilities and fixity of initial decisions is modeled. The  $X$  variables are fixed across all second stage states of nature, but the  $Z$  variables adapt to the state of nature.

Third, let us examine the linkages between the stages. The coefficients reflect a potentially risky link between the predecessor ( $X$ ) and successor ( $Z$ ) activities through the third constraint. Note the link is essential since if the activities are not linked, then the problem is not a sequential decision problem. These links may involve the weighted sum of a number of predecessor and successor variables (i.e., an uncertain quantity of lumber harvested via several cutting schemes linked with use in several products). Also, multiple links may be present (i.e., there may be numerous types of lumber). The subscript  $m$  defines these links. A fourth comment relates to the nature of uncertainty resolution. The formulation places all uncertainty into the objective function, which maximizes expected income.

#### 14.4.1.1 Example

Let us consider a simple farm planning problem. Suppose we can raise corn and wheat on a 100 acre farm. Suppose per acre planting cost for corn is \$100 while wheat costs \$60. However, suppose crop yields, harvest time requirements per unit of yield, harvest time availability and crop prices are uncertain. The deterministic problem is formulated as in Table 14.20 and file SPREXAM1. Here the harvest activities are expressed on a per unit yield basis and the income variable equals sales revenue minus production costs.

The uncertainty in the problem is assumed to fall into two states of nature and is expressed in Table 14.19. These data give a joint distribution of all the uncertain parameters. Here RHS's,  $a_{ij}$ 's and objective function coefficient's are uncertain.

Solution of the Table 14.20 LP formulation under each of the states of nature gives two very different answers. Namely under the first state of nature all acreage is in corn while under the second state of nature all production is in wheat. These are clearly not robust solutions.

The SPR formulation of this example is given in Table 14.21. This tableau contains one set of first stage variables (i.e., one set of corn growing and wheat growing activities) coupled with two sets of second stage variables after the uncertainty is resolved (i.e., there are income, harvest corn, and harvest wheat variables for both states of nature). Further, there is a single unifying objective function and land constraint, but two sets of constraints for the states of nature (i.e., two sets of corn and wheat yield balances, harvesting hour constraints and income constraints). Notice underneath the first stage corn and

wheat production variables, that there are coefficients in both the state of nature dependent constraints reflecting the different uncertain yields from the first stage (i.e., corn yields 100 bushels under the first state of nature and 105 under the second; while wheat yields 40 under the first and 38 under the second). However, in the second stage resource usage for harvesting is independent. Thus, the 122 hours available under the first state of nature cannot be utilized by any of the activities under the second state of nature. Also, the crop prices under the harvest activities vary by state of nature as do the harvest time resource usages.

The example model then reflects, for example, if one acre of corn is grown that 100 bushels will be available for harvesting under state of nature one, while 105 will be available under state of nature two. In the optimum solution there are two harvesting solutions, but one production solution. Thus, we model irreversibility (i.e., the corn and wheat growing variable levels maximize expected income across the states of nature, but the harvesting variable levels depend on state of nature).

The SPR solution to this example is shown in Table 14.22. Here the acreage is basically split 50-50 between corn and wheat, but harvesting differs with almost 4900 bushels of corn harvested under the first state, where as 5100 bushels of corn are harvested under the second. This shows adaptive decision making with the harvest decision conditional on state of nature. The model also shows different income levels by state of nature with \$18,145 made under state of nature one and \$13,972 under state of nature two. Furthermore, note that the shadow prices are the marginal values of the resources times the probability of the state of nature. Thus, wheat is worth \$3.00 under the first state of nature but taking into account that the probability of the first state of nature is 60% we divide the \$3.00 by .6 we get the original \$5.00 price. This shows the shadow prices give the contribution to the average objective function. If one wishes shadow prices relevant to income under a state of nature then one needs to divide by the appropriate probability.

The income accounting feature also merits discussion. Note that the full cost of growing corn is accounted for under both the first and second states of nature. However, since income under the first state of nature is multiplied by .6 and income under the second state of nature is multiplied by .4, then no double counting is present.

#### **14.4.2 Incorporating Risk Aversion**

The two stage model as presented above is risk neutral. This two stage formulation can be altered to incorporate risk aversion by adding two new sets of constraints and three sets of variables following the method used in the unified model above. An EV formulation is

$$\begin{array}{ll}
\text{Max} & \bar{Y} \\
& - R \sum_k p_k (d_k^+ + d_k^-)^2 \\
\text{s.t.} & \sum_k p_k Y_k = 0 \\
& \sum_k Y_k + d_k^+ - d_k^- = 0 \quad \text{for all } k \\
& \sum_j c_j X_j + \sum_n e_{nk} Z_{nk} = 0 \quad \text{for all } k \\
& \sum_j a_{ij} X_j = b_i \quad \text{for all } i \\
& \sum_j d_{mjk} X_j + \sum_n f_{mnk} Z_{nk} = 0 \quad \text{for all } m,k \\
& \sum_n g_{wnk} Z_{nk} = s_{wk} \quad \text{for all } w,k \\
& \bar{Y}, Y_k \leq 0 \quad \text{for all } k \\
& X_j, d_k^+, d_k^-, Z_{nk} \geq 0 \quad \text{for all } j,n,k
\end{array}$$

Note that within this formulation the first new constraint that we add simply accounts expected income into a variable  $\bar{Y}$ , while the second constraint computes deviations from expected income into new deviation variables  $d_k^+$ ,  $d_k^-$  which are defined by state of nature. Further, the objective function is modified so it contains expected income minus a risk aversion parameter times the probabilistically weighted squared deviations (i.e., variance). This is as an EV model. The model may also be formulated in the fashion of the unified model discussed earlier to yield either a MOTAD or an E-standard deviation model.

#### 14.4.2.1 Example

Suppose we use the data from the above Wicks Guise example but also allow decision makers once they discover the state of nature, to supplement the diet. In this case, suppose the diet supplement to correct for excess protein deviation costs the firm \$0.50 per protein unit while insufficient protein costs \$1.50 per unit. Similarly, suppose excess energy costs \$1.00 per unit while insufficient energy costs \$0.10. The resultant SPR tableau, portraying just two of the four states of nature included in the tableau, is shown in Table 14.23 (This smaller portrayal is only done to preserve readability, the full problem is solved). Notice we again have the standard structure of an SPR. Namely the corn, soybeans, and wheat activities are first stage activities, then in the second stage there are positive and negative nutrient deviations for each state as well as state dependent objective function and deviation variable accounting. Notice the average cost row adds the probabilistically weighted sums of the state of nature dependent variables into average cost while the cost deviation rows compute deviation under a particular state of nature. In turn, these deviations are weighted by the probability times the risk aversion parameter and are entered in the objective function. The deviation variables could be treated to form an E-V, MOTAD or E-Standard error formulation as in the unified model above. An E-standard deviation model will be used here and is implemented in the GAMS file FEEDSPR. Also note these activities repeat for the second state of nature and also would for the third and fourth if they were portrayed here.

The risk neutral solution to this problem is given in Table 14.24. Two solution aspects are worth discussing. First, notice that the first stage solution is to buy .283 pounds of corn, .362 pounds of soybeans, .355 pounds of wheat at an average cost of 6.7 cents. Cost varies across the states of nature with cost under the first state equaling 8.1 cents, while under the second state it is 8.3, 5.2 under the third state and 5.1 under the fourth state. The cost variation arises as the protein and energy shortfall and excess variables take on different values in order to mitigate nutrient fluctuation.

The model was also solved for risk aversion. The results in Table 14.25 show the solutions from the example model under varying risk aversion coefficients for a standard deviation implementation. Table 14.25 gives the changes in corn, soybean, and wheat usage, as well as average income and standard error of income as the risk aversion parameter is changed for an E-standard deviation formulation as implemented in the file FEEDSPR. Here the risk aversion parameter was varied from 0.0 up to 0.6. As risk aversion increases the average cost of the diet increases, but the standard error of the cost of the diet falls with cost variation between the various states of nature narrowing. Namely under risk neutrality cost ranges from 5.1 cents to 8.1 cents with a standard error of 1.5 cents, however by the time the risk aversion parameter is up to .4 the cost varies from only 6.7 to 7.4 cents with a standard error of two tenths of a cent, at the expense of a 0.4 cent increase in average diet cost. Thus, as risk aversion increases, the model adopts a plan which stabilizes income across all of the states of nature.

#### **14.4.3 Extending to Multiple Stages**

The models above are two stage models with a set of predecessor activities followed by sets of successor activities for each state of nature. It is possible to formulate a multiple stage model as done by Cocks. In such a model however, it is relatively cumbersome to express a general formulation. Thus, we will express this model only in terms of an example (See Cocks for an N stage formulation and Boisvert and McCarl for a three stage one). Let us formulate a relatively simple stock model. Suppose that a firm starts with an initial inventory 100 units of common stock and is trying to maximize average ending worth. In doing this, suppose that the stock can be sold in one of three time periods. The first one which is nonstochastic, the second one which is characterized by two states of nature, and the third which is characterized by two additional states of nature. In describing the states of nature the following data are relevant. In period one (today) the firm knows the price is \$2.00. In period two, the firm is uncertain of the interest rate between periods and the future price. Assume that under state of nature 1, the interest rate between period one and two for any stock sold is one percent while it is two percent under the second state of nature 2. Simultaneously the stock price is \$2.20 under the first state of nature and \$2.05 under the second. Going into the third state of nature, the interest rate is conditional on which state of nature was drawn for the second state. Thus, in the third stage if the first state arose the third stage interest rates are then either 6% (A) or 4% (B). On the other hand if the second state occurs, the interest rate will either be

7% (A) or 3% (B). Third stage crop prices are dependent of which of the two third stage states of nature occur. Under the first state of nature (A) the price is \$2.18, while under the second one it is \$2.44. The third stage probabilities are also conditional. Namely, after the first stage one gets state 1 occurring 70% of the time while state 2 occurs 30% of the time. When state 2 results out of stage one then the third stage probability for state A is 60% and is 40% for state B. On the other hand, these probabilities change to .7 and .3 if the second state happened out of stage 1.

The resultant formulation of this problem is given in Table 14.26 and file SELLSPR. Here, again, there is one set of period one variables which refer to either keeping or selling the stock; two sets of period two variables, which refer again to keep or sell the stock under each second stage state of nature; and four sets of period three variables for selling the stock and accounting ending net worth under all the third stage states of nature. Note in the first period, if the stock is kept, it carries over from the first period to both states of nature in the second stage. Then in the second period the keep activity from the first period provides stock that could either be sold or kept on into the third. In turn, if stock is kept in the second stage, it is held over to both third period states of nature which follow that second period state of nature. Notice the probabilities of each of the final states are reflected in the average ending worth. The worth under period three state A following period two state one is multiplied 0.42 which reflects the 70% probability of period two state one times the 60% conditional probability of period three state A. Also, notice the prices as they enter the ending worth by state of nature are the sales price in the relevant period times 1 plus interest earned in the interim periods. Thus, the ending worth of period one sales following period two state one and period three state A is 2.1412. This reflects the original sales price of \$2.00, the 1% interest into the second period and the 6% interest into the third period. The solution to this model is given in Table 14.27.

#### **14.4.4 Model Discussion**

The SPR model is perhaps the most satisfying of the risk models. Conceptually it incorporates all sources of uncertainty: right hand side, objective function and technical coefficients while allowing adaptive decisions. However, the formulations suffer from the "curse of dimensionality." Each possible final state of nature leads to another set of stage two or later activities and large models can result from relatively simple problems. For example, consider having ten values of two right hand sides which were independently distributed. This would lead to 100 terminal states or sets of rows. However, such models can be computationally tractable, since the sparsity and repeated structure tend to make such problems easier to solve than their size would imply. Thus, one of the things to be cautious about when using this particular formulation is size. When dealing with such a model, it is often advisable to determine the critical sources of uncertainty which should be extensively modeled. Uncertainties other than the "most critical" may be handled with such methods as MOTAD, Wicks and Guise, or chance-constrained as

discussed above. Sources of uncertainty which are not important in the problem may be held at their expected values (see Tice for an example). Thus, with careful application, this type of model can be quite useful.

Agricultural economics applications include Yaron and Horowitz (1972a); Garoian, et al.; Apland, et al. (1981); Lambert and McCarl (1989); Leatham and Baker; McCarl and Parandvash; and the early papers by Rae (1971a, 1971b). Hansotia; Boisvert and McCarl; and Apland and Kaiser provide literature reviews.

#### **14.5 General Comments On Modeling Uncertainty**

As demonstrated above, there are a number of ways of handling uncertainty when modeling. Several aspects of these types of models need to be pointed out. First, all the formulations convert the problems to a deterministic equivalent. Basically, it is assumed that the decision maker is certain of the risk and reacts to it optimally by discounting the objective function,  $a_{ij}$ s or right hand sides. Obviously this means the modeler must assume knowledge of the distribution of risk faced by a decision maker and the risk aversion coefficient.

The second set of comments regards data. Important parameters within the context of risk models are the expectation of the coefficient value and its probability distribution around that expectation. The most common practice for specification of these parameters is to use the historical mean and variance. This, however, is neither necessary nor always desirable. Fundamentally, the measures that are needed are the value expected for each uncertain parameter and the perceived probability distribution of deviations from that expectation (with joint distributions among the various uncertain parameters). The parameter expectation is not always a historical mean. This is most unrealistic in cases where there has been a strong historical trend (as pointed out by Chen, 1971). There is a large body of literature dealing with expectations and/or time series analysis (see Judge for an introduction), and some use of these results and procedures appears desirable.

Data are most often generated historically; however, observations could be generated by several other means. For example, observations could be developed from a simulation model (see Dillon, et al.), from a forecasting equation (see Lambert and McCarl (1989)), or from subjective interrogation of the decision maker (see Sri Ramaratnam et al.). There are cases where these other methods are more appropriate than history due to such factors as limited historical data (say, on the price of a new product) or major structural changes in markets. Naturally, the form in which the data are collected depends on the particular application involved.

A final comment on data regards their probabilistic nature. Basically when using historically based means and variance one is assuming that all observations are equally probable. When this assumption is invalid, the model is modified so that the value expected is the probabilistically weighted mean (if desired)



and the variance formula includes the consideration of probability (see Anderson, et al. [pp. 28- 29] for examples). Deviation models must also be adjusted so that the deviations are weighted by their probability as done in the MOTAD version of the discrete stochastic model in section 14.23.

A third and again independent line of comment relates to the question "should uncertainty be modeled and if so, how?" Such a concern is paramount to this section. It is obvious from the above that in modeling uncertainty, data are needed describing the uncertainty, and that modeling uncertainty makes a model larger and more complex, and therefore harder to interpret, explain, and deal with. It is not the purpose of these comments to resolve this question, but rather to enter some considerations to the resolution of this question. First and fundamentally, if a model solution diverges from reality because the decision maker in reality has somehow considered risk, then it is important to consider risk. This leads to the subjective judgment on behalf of the modeling team as to whether risk makes a difference. Given that risk is felt to make a difference, then, how should risk be modeled? In the approaches above, the formulation model depends upon whether there is conditional decision making and on what is uncertain. These formulations are not mutually exclusive; rather, it may be desirable to use combinations of these formulations (see, for example, Wicks and Guise, Tice or Klemme).

Several uncertainty models have not been covered the above discussion. There are more advanced applications of chance constrained programming such as those found in the books by Sengupta; Vajda; and Kolbin. Another approach is called "Cautions Suboptimizing" by Day (1979). This approach bounds the adjustments in variables to a maximum amount in any one year. We also have not covered Monte Carlo programming as espoused by Anderson, et al., mainly because we do not feel it falls into the class of programming techniques but rather is a simulation technique.

Finally, it is relevant to discuss how risk should be modeled. There have been arguments presented in literature (e.g. see, for example, Baker and McCarl or Musser, et al.) that risk model solutions are biased if the model structure is not adequate before risk modeling is incorporated. Baker and McCarl argue that one should not include risk until the model structure is fully specified in terms of the needed constraints, the time disaggregation of constraints, and activities.

## References

- Agrawal, R.C. and E.O. Heady. "Application of Game Theory in Agriculture." Journal of Agricultural Economics. 19(1968):207-218.
- Anderson, J.R., J.L. Dillon and J.B. Hardaker. Agricultural Decision Analysis. Ames, Iowa: The Iowa State University Press, 1977.
- Apland, J.D., B.A. McCarl, and T. Baker. "Crop Residue for Energy Generation: A Prototype Application to Midwestern USA Grain Prices." Energy in Agriculture. 1(1981):55-78.
- Apland, J.D., B.A. McCarl, and W.L. Miller. "Risk and the Demand for Supplemental Irrigation: A Case Study in the Corn Belt." American Journal of Agricultural Economics. 62(1980):142-145.
- Apland, J.D. and H. Kaiser. "Discrete Stochastic Sequential Programming: A Primer." Staff Papers Series, Paper P84-8. Institute of Agriculture, Forestry and Home Economics. St. Paul: University of Minnesota, 1984.
- Baker, T.G. and B.A. McCarl. "Representing Farm Resource Availability Over Time in Linear Programs: A Case Study." North Central Journal of Agricultural Economics. 4(1982):59-68.
- Barry, P. J. (ed.) Risk Management in Agriculture. Ames, Iowa: Iowa State University Press, 1984.
- Blau, R.A. "Stochastic Programming and Decision Analysis: An Apparent Dilemma." Management Science. 21(1974):271-276.
- Boisvert, R. "Available Field Time, Yield Losses and Farm Planning." Canadian Journal of Agricultural Economics. 24(1976):21-32.
- Boisvert, R.N. and H. Jensen. "A Method for Planning Under Uncertain Weather Conditions, with Applications to Corn-Soybean Farming in southern Minnesota." University of Minnesota Agricultural Experiment Station Tech. Bulletin No. 292, 1973.
- Boisvert, R.N. and B.A. McCarl. Agricultural Risk Modeling Using Mathematical Programming. Southern Cooperative Series Bulletin No. 356. Cornell University, New York. July 1990.
- Brink, L. and B.A. McCarl. "The Adequacy of a Crop Planning Model for Determining Income, Income Change, and Crop Mix." Canadian Journal of Agricultural Economics. 27(1979):13-15.
- Brainard, W.C. and R.N. Cooper. "Uncertainty and Diversification in International Trade." Federal Reserve Institute Studies. 8(1968):257-85.
- Buccola, S.T. "Minimizing Mean Absolute Deviations to Exactly Solve Expected Utility Problems: Comment." American Journal of Agricultural Economics. 64(1982):789-91.
- Bussey, L.E. The Economic Analysis of Industrial Projects. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1978.
- Candler, W. and M. Boeljhe. "Use of Linear Programming in Capital Budgeting with Multiple Goals." American Journal of Agricultural Economics. 53(1977):325-330.
- Charnes, A. and W.W. Cooper. "Chance Constrained Programming." Management Science. 6(1959):73-79.
- Chen, J.T. "Quadratic Programming for Least-Cost Feed Formulations Under Probabilistic Protein Constraints." American Journal of Agricultural Economics. 55(1973):175-183.
- Chen, J.T. "A Linear Alternative to Quadratic and semivariance Programming for Farm Planning Under Uncertainty: Comment." American Journal of Agricultural Economics. 53(1971):662-663.
- Chen, J.T. and C.B. Baker. "Marginal Risk Constraint Linear Program for Activity Analysis." American Journal of Agricultural Economics. 56(1976):456-465.
- Cocks, K.D. "Discrete Stochastic Programming." Management Science. 15(1968):72-79.

- Collender, R.N. and J.A. Chalfant. "An Alternative Approach to Decisions Under Uncertainty Using the Empirical Moment-Generating Function." American Journal of Agricultural Economics. 68(1986):727-731.
- Collender, R.N. and D. Zilberman. "Land Allocation Under Uncertainty for Alternative Technologies With Stochastic Yield." American Journal of Agricultural Economics. 67(1985):779-793.
- Curtis, C.E., G.H. Pfeiffer, L.L. Lutgen and S.D. Frank. "A Target MOTAD Approach to Marketing Strategy Selection for Soybeans." North Central Journal of Agricultural Economics. 9(1987):195-206.
- Danok, A.B., B.A. McCarl and T.K. White. "Machinery Selection Modeling: Incorporation of Weather Variability." American Journal of Agricultural Economics. 62(1980):700-08.
- Dantzig, G. "Linear Programming Under Uncertainty." Management Science. 1(1955):197-206.
- Day, R. "Cautious Suboptimizing." Risk, Uncertainty and Agricultural Development. Editors J. Roumasset, J. Boussard and I. Singhe, Agricultural Development Council, New York.
- Dillon, C.R., J.W. Mjelde and B.A. McCarl. "Biophysical Simulation in Support of Crop Production Decisions: A Case Study in the Blacklands Region of Texas." Southern Journal of Agricultural Economics. 21(1989):73-86.
- Dubman, R.W., L. F. Gunter and B.R. Miller. "Revenue and Cost Uncertainty, Generalized Mean-Variance and the Linear Complementarity Problem: Comment." American Journal of Agricultural Economics. 61(1989):806-809.
- Eisel, L. "Chance Constrained Reservoir Model." Water Resources Research. 8:(1972):339-347.
- Featherstone, A.M., C.B. Moss, T.G. Baker and P.V. Preckel. "The Theoretical Effects of Farm Policies on Optimal Leverage and the Probability of Equity Losses." American Journal of Agricultural Economics. 70(1988):572-579.
- Frank, S.D., S.H. Irwin, G.H. Pfeiffer, and C.E. Curtis. "Further Evidence on Soybean Marketing Strategies: The Role of Options." North Central Journal of Agricultural Economics. 11(1989):213-220.
- Freund, R. "The Introduction of Risk into a Programming Model." Econometrica. 21(1956):253-263.
- Garoian, L., J.R. Conner and C.J. Scifres. "A Discrete Stochastic Programming Model to Estimate Optimal Burning Schedules on Rangeland." Southern Journal of Agricultural Economics. 19(1987):53-60.
- Gebremeskel, T. and C.R. Shumway. "Farm Planning and Calf Marketing Strategies for Risk Management: An Application of Linear Programming and Statistical Decision Theory." American Journal of Agricultural Economics. 61(1979):363-370.
- Hansotia, B.J. "Stochastic Linear Programming With Recourse: A Tutorial." Decision Sciences. 11(1980):151-168.
- Hazell, P.B.R. "Game Theory - An Extension of Its Application to Farm Planning Under Uncertainty." Journal of Agricultural Economics. 21(1970):239-252.
- Hazell, P.B.R. "A Linear Alternative to Quadratic and Semivariance Programming for Farm Planning under Uncertainty." American Journal of Agricultural Economics. 53(1971):53-62.
- Hazell, P.B.R. and R.B. How. "Obtaining Acceptable Farm Plans Under Uncertainty." Papers and Reports 14th International Conference of Agricultural Economists. pp. 338-47. Oxford: Institute for Agricultural Affairs. 1971.
- Hazell, P.B.R. and P.L.Scandizzo. "Market Intervention Policies when Production is Risky." American Journal of Agricultural Economics. 57(1975):641-649.
- Hazell, P.B.R., R.D. Norton, M. Parthasarthy and C. Pomereda. "The Importance of Risk in Agricultural

- Planning Models." The Book of CHAC: Programming Studies for Mexican Agriculture. R.D. Norton and L. Solis (eds.), Baltimore, Md: Johns Hopkins University Press, 1983.
- Heady, E.O. and W. Candler. Linear Programming Methods. Ames: Iowa State University Press, 1958.
- Hogan, A.J., J.G. Morris and H.E. Thompson. "Decision Problems Under Risk and Chance Constrained Programming: Dilemmas in the Transition." Management Science. 27(1981):716.
- Hogg, R.V. and A.T. Craig. Introduction to Mathematical Statistics. New York: Macmillan Company, 1970.
- Jabara, C.L. and R.L. Thompson. "Agricultural Comparative Advantage Under International Price Uncertainty: The Case of Senegal." American Journal of Agricultural Economics. 62(1980):188-198.
- Johnson, D.A. and M.D. Boehlje. "Managing Risk by Coordinating Investment, Marketing and Production Strategies." Western Journal of Agricultural Economics. 6(1983):155-169.
- Johnson, D. and M.D. Boehlje. "Minimizing Mean Standard Deviations to Exactly Solve Expected Utility Problems." American Journal of Agricultural Economics. 63(1981):728-29.
- Judge, G.G. "Estimation and Statistical Inference in Economics." A Survey of American Agricultural Economics Literature. Vol. 2., L.R. Martin, ed. Minneapolis: University of Minnesota Press, (1977):3-50.
- Kawaguchi, T. and Y. Maruyama. "Generalized Constrained Games in Farm Planning." American Journal of Agricultural Economics. 54(1972):591-702.
- Kaylen, M.S., R.V. Preckel and E.T. Loehman. "Risk Modeling Via Direct Utility Maximization Using Numerical Quadrature." American Journal of Agricultural Economics. 69(1987):701-706.
- Klemme, R.M. "An Economic Analysis of the On-farm Grain Handling Decision Problem." Unpublished Ph.D. Dissertation, Purdue University, May 1980.
- Kolbin, V.V. Stochastic Programming. Boston:D. Reidel Publishing Co. 1977.
- Kroll, Y., H. Levy, and H.M. Markowitz. "Mean-Variance Versus Direct Utility Maximization." Journal of Finance. 59(1984):47-62.
- Lambert, D.K. "Calf Retention and Production Decisions Over Time." Western Journal of Agricultural Economics. 14(1989):9-19.
- Lambert, D.K. and B.A. McCarl. "Sequential Modeling of White Wheat Marketing Strategies." North Central Journal of Agricultural Economics. 11(1989):105-115.
- Lambert, D. and B.A. McCarl. "Risk Modeling Using Direct Solution of Nonlinear Approximations of the Utility Function." American Journal of Agricultural Economics. 67(1985):846-852.
- Leatham, D.J. and T.G. Baker. "Farmers' Choice of Fixed and Adjustable Interest Rate Loans." American Journal of Agricultural Economics. 70(1988):803-812.
- Levy, H. and H. Markowitz. "Approximating Expected Utility by a Function of Mean and Variance". American Economic Review. 69(1979):308-317.
- Lin, W., G.W. Dean and C.V. Moore. "An Empirical Test of Utility vs. Profit Maximization in Agricultural Production." American Journal of Agricultural Economics. 56(1974):497-508.
- Loucks, D. "An Evaluation of Some Linear Decision Rules in Chance Constrained Models for Reservoir Planning and Operation." Water Resource Research. 11(1975):777-82.
- Low, A.R.C. "Decision Making Under Uncertainty: A Linear Programming Model of Peasant Farmer Behavior." Journal of Agricultural Economics. 25(1974):311-322.
- Maji, C. and E. Heady. "Intertemporal Allocation of Irrigation Water in the Mayurakshi Project (India): An Application of Chance Constrained Linear Programming." Water Resources Research. 14(1978):190-205.

- Mapp, H.P. Jr., M.L. Hardin, O.L. Walker and T. Persand. "Analysis of Risk Management Strategies for Agricultural Producers." American Journal of Agricultural Economics. 61(1979):1071-1077.
- Markowitz, H.M. Portfolio Selection: Efficient Diversification of Investments. New York: John Wiley and Sons, Inc., 1959.
- Maruyama, Y. "A Truncated Maximum Approach to Farm Planning Under Uncertainty with Discrete Probability Distributions." American Journal of Agricultural Economics. 54(1972):192-200.
- McCamley, F. and J.B. Kliebenstein. "Describing and Identifying the Complete Set of Target MOTAD Solutions." American Journal of Agricultural Economics. 69(1987):669-76.
- McCarl, B.A. and D. Bessler. "Estimating an Upper Bound on the Pratt Risk Aversion Coefficient When the Utility Function is Unknown." Australian Journal of Agricultural Economics. 33(1989):56-63.
- McCarl, B.A., W. Candler, D. Doster, and P. Robbins. "Experiences with Farmer Oriented Linear Programming for Crop Farming." Canadian Journal of Agricultural Economics. 24(1977):17-30.
- McCarl, B.A. and H. Onal. "Linear Approximation of Using MOTAD and Separable Programming: Should It Be Done." American Journal of Agricultural Economics. 71(1989):158-165.
- McCarl, B.A. and G.H. Parandvash. "Irrigation Develop Versus Hydroelectric Generation: Can Interruptable Irrigation Play a Role." Western Journal of Agricultural Economics. 13(1988):267-276.
- McCarl, B.A. and T. Tice. "Should Quadratic Programming Problems be Approximated?" American Journal of Agricultural Economics. 64(1982):585-589.
- McFarquhar, A.M.M. "Rational Decision Making and Risk in Farm Planning - An Application of Quadratic Programming in British Arable Farming." Journal of Agricultural Economics. 14(1961):552-563.
- McInerney, J.P. "Maximum Programming - An Approach to Farm Planning Under Uncertainty." Journal of Agricultural Economics. 18(1967):279-290.
- McInerney, J.P. "Linear Programming and Game Theory Models - Some Extensions." Journal of Agricultural Economics. 20(1969):269-278.
- Merrill, W.C. "Alternative Programming Models Involving Uncertainty." Journal of Farm Economics. 47(1965):595-610.
- Meyer, J. "Two-Moment Decision Models and Expected Utility Maximization." American Economic Review. 77(1987):421-430.
- Moffit, L.J., T.M. Burrows, J.L. Baritelle and V. Sevacherian. "Risk Evaluation of Early Termination for pest Control in Cotton." Western Journal of Agricultural Economics. 9(1984):145-151.
- Musser, W.N., B.A. McCarl and G.S. Smith. "An Investigation of the Relationship Between Constraint Omission and Risk Aversion in Firm Risk Programming Models." Southern Journal of Agricultural Economics. 18(1986):147-154.
- Nieuwoudt, W.L., J.B. Bullock and G.A. Mathia. "An Economic Evaluation of Alternative Peanut Policies." American Journal of Agricultural Economics. 58(1976):485-495.
- Paris, Q. "Revenue and Cost Uncertainty, Generalized Mean-Variance, and the Linear Complementarity Problem." American Journal of Agricultural Economics. 61(1979):268-275.
- Paris, Q. "Revenue and Cost Uncertainty, Generalized Mean-Variance and the Linear Complementarity Problem: Reply." American Journal of Agricultural Economics. 61(1989):810-812.
- Pfaffenberger, R.C. and D.A. Walker. Mathematical Programming for Economics and Business. Ames: Iowa State University Press, 1976.
- Pomareda, C. and O. Samayoa. "Area and Yield Response to Price Policy: A Case Study in Guatemala, C.A." American Journal of Agricultural Economics. 61(1979):683-86.
- Preckel, P.V., A.M. Featherstone, and T.G. Baker. "Interpreting Dual Variables for Optimization with

- Nonmonetary Objectives." American Journal of Agricultural Economics. 69(1987):849-851.
- Rae, A.N. "Stochastic Programming, Utility, and Sequential Decision Problems in Farm Management." American Journal of Agricultural Economics. 53(1971a):448-60.
- Rae, A.N. "An Empirical Application and Evaluation of Discrete Stochastic Programming in Farm Management." American Journal of Agricultural Economics. 53(1971b):625-38.
- Rahman, S.A. and F.E. Bender. "Linear Programming Approximation of Least-Cost Feed Mixes with Probability Restrictions." American Journal of Agricultural Economics. 53(1971):612-18.
- Reid, D.W. and B.V. Tew. "An Evaluation of Expected Value and Expected Value-Variance Criteria in Achieving Risk Efficiency in Crop Selection." Northeastern Journal of Agricultural and Resource Economics. 16(1987):93-101.
- Robinson, L.J. and J.R. Brake. "Application of Portfolio Theory to Farmer and Lender Behavior." American Journal of Agricultural Economics. 61(1979):159-164.
- Roy, A.D. "Safety-First and the Holding of Assets." Econometrica. 20(1952):431-449.
- Schurle, B.W. and B.L. Erven. "Sensitivity of Efficient Frontiers Developed for Farm Enterprise Choice Decisions." American Journal of Agricultural Economics. 61(1979):506-511.
- Sengupta, J.K. Stochastic Programming: Methods and Applications. New York: American Elsevier Publishing Company, Inc. 1972.
- Sharpe, W. "A Linear Programming Algorithm for Mutual Fund Portfolio Selection." Management Science. 14(1967):499-510.
- Simmons, R.L. and C. Pomareda. "Equilibrium Quantity and Timing of Mexican Vegetable Exports." American Journal of Agricultural Economics. 57(1975):472-479.
- Sri Ramaratnam, S., M.E. Rister, D.A. Bessler, and James Novak. "Risk Attitudes and Farm/Producer Attributes: A Case Study of Texas Coastal Bend Grain Sorghum Producers." Southern Journal of Agricultural Economics. 18(1986):85-95.
- Stovall, J. "Income Variation and Selection of Enterprises." Journal of Farm Economics. 48(1966):1575-1579.
- Tauer, L. "Target MOTAD." American Journal of Agricultural Economics. 65(1983):606-610.
- Thomas, W., L. Blakeslee, L. Rogers and N. Whittlesey. "Separable Programming for Considering Risk in Farm Planning." American Journal of Agricultural Economics. 54(1972):260-266.
- Thomson, K. and P.B.R. Hazell. "Reliability of Using the Mean Absolute Deviation to Derive Efficient E-V farm Plans." American Journal of Agricultural Economics. 54(1972):503-506.
- Tice, T.F. "An Investigation of Nitrogen Application Under Uncertainty." Unpublished Ph.D. Dissertation, Purdue University, August 1979.
- Tobin, J. "Liquidity Preference as Behavior toward Risk." Review of Economic Studies. 25(1958):65-86.
- Townsley, R. "Derivation of Optimal Livestock Rations Using Quadratic Programming." Journal of Agricultural Economics. 19(1968):347-354.
- Tsiang, S. "The Rationale of the Mean-Standard Deviation Analysis, Skewness, Preference and the Demand for Money." American Economic Review. 62(1972):354-371.
- Tsiang, S. "The Rationale of the Mean-Standard Deviation Analysis: Reply and Errata for Original Article." American Economic Review. 64(1974):442-450.
- Vajda, S. Probabilistic Programming. New York: Academic Press, 1972.
- von Neumann, J. and O. Morgenstern. Theory of Games and Economic Behavior. Princeton, N.J.: Princeton University Press, 1947.

- Wagner, H.M. Principles of Operations Research. Englewood Cliffs, N.J.: Prentice Hall, Inc. 1975.
- Weins, J.A. "Peasant Risk Aversion and Allocative Behavior: A Quadratic Programming Experiment." American Journal of Agricultural Economics. 58(1976):629-635.
- Wicks, J.A. and J.W.B. Guise. "An Alternative Solution to Linear programming Problems with Stochastic Input-Output Coefficients." Australian Journal of Agricultural Economics. 22(1978):22-40.
- Yaron, D., and V. Horowitz. "A Sequential Programming Model of Growth and Capital Accumulation of a Farm Under Uncertainty." American Journal of Agricultural Economics. 54(1972):441-451.
- Yassour, J.D., D. Zilberman and G.C. Rausser. "Optional Choices Among Alternative Technologies with Stochastic Yield." American Journal of Agricultural Economics. 63(1981):718-723.
- Yitzhaki, S. "Stochastic Dominance, Mean Variance and Gini's Mean Difference." American Economic Review. 72(1982):178-185.
- Young, D.L. "Evaluating Procedures for Computing Objective Risk from Historical Time Series." Paper presented at Annual Meeting of Western Regional Research Project W-149, Tucson, Arizona. January 16-18, 1980.
- Zimet, D.J. and T.H. Spreen. "A Target MOTAD Analysis of a Crop and Livestock Farm in Jefferson County, Florida." Southern Journal of Agricultural Economics. 18(1986):175-185.

**Table 14.1. Data for E-V Example -- Returns by Stock and Event**

----Stock Returns by Stock and Event----				
	Stock1	Stock2	Stock3	Stock4
Event1	7	6	8	5
Event2	8	4	16	6
Event3	4	8	14	6
Event4	5	9	-2	7
Event5	6	7	13	6
Event6	3	10	11	5
Event7	2	12	-2	6
Event8	5	4	18	6
Event9	4	7	12	5
Event10	3	9	-5	6
	Stock1	Stock2	Stock3	Stock4
Price	22	30	28	26

**Table 14.2. Mean Returns and Variance Parameters for Stock Example**

	Stock1	Stock2	Stock3	Stock4
Mean Returns	4.70	7.60	8.30	5.80
Variance-Covariance Matrix				
	Stock1	Stock2	Stock3	Stock4
Stock1	3.21	-3.52	6.99	0.04
Stock2	-3.52	5.84	-13.68	0.12
Stock3	6.99	-13.68	61.81	-1.64
Stock4	0.04	0.12	-1.64	0.36

**Table 14.3. GAMS Formulation of E-V Problem**

```

5 SETS      STOCKS  POTENTIAL INVESTMENTS / BUYSTOCK1*BUYSTOCK4 /
6          EVENTS  EQUALLY LIKELY RETURN STATES OF NATURE
7
8                               / EVENT1*EVENT10 / ;
9 ALIAS (STOCKS,STOCK);
10
11 PARAMETERS  PRICES(STOCKS) PURCHASE PRICES OF THE STOCKS
12
13                / BUYSTOCK1  22
14                BUYSTOCK2  30
15                BUYSTOCK3  28
16                BUYSTOCK4  26 / ;

```



```

16
17 SCALAR FUNDS TOTAL INVESTABLE FUNDS / 500 / ;
18
19 TABLE RETURNS(EVENTS,STOCKS) RETURNS BY STATE OF NATURE EVENT
20
21 BUYSTOCK1 BUYSTOCK2 BUYSTOCK3 BUYSTOCK4
22 EVENT1 7 6 8 5
23 EVENT2 8 4 16 6
24 EVENT3 4 8 14 6
25 EVENT4 5 9 -2 7
26 EVENT5 6 7 13 6
27 EVENT6 3 10 11 5
28 EVENT7 2 12 -2 6
29 EVENT8 5 4 18 6
30 EVENT9 4 7 12 5
31 EVENT10 3 9 -5 6
32
33 PARAMETERS
34 MEAN (STOCKS) MEAN RETURNS TO X(STOCKS)
35 COVAR(STOCK,STOCKS) VARIANCE COVARIANCE MATRIX;
36
37 MEAN(STOCKS) = SUM(EVENTS , RETURNS(EVENTS,STOCKS) / CARD(EVENTS) );
38 COVAR(STOCK,STOCKS)
39 = SUM (EVENTS ,(RETURNS(EVENTS,STOCKS) - MEAN(STOCKS))
40 *(RETURNS(EVENTS,STOCK) - MEAN(STOCK)))/CARD(EVENTS);
41
42 DISPLAY MEAN , COVAR ;
43
44 SCALAR RAP RISK AVERSION PARAMETER / 0.0 / ;
45
46 POSITIVE VARIABLES INVEST(STOCKS) MONEY INVESTED IN EACH STOCK
47
48 VARIABLE OBJ NUMBER TO BE MAXIMIZED ;
49
50 EQUATIONS OBJJ OBJECTIVE FUNCTION
51 INVESTAV INVESTMENT FUNDS AVAILABLE
52 ;
53
54 OBJJ..
55 OBJ =E= SUM(STOCKS, MEAN(STOCKS) * INVEST(STOCKS))
56 - RAP*(SUM(STOCK, SUM(STOCKS,
57 INVEST(STOCK)* COVAR(STOCK,STOCKS) * INVEST(STOCKS)));
58
59 INVESTAV.. SUM(STOCKS, PRICES(STOCKS) * INVEST(STOCKS)) =L= FUNDS ;
60
61 MODEL EVPORTFOL /ALL/ ;
62
63 SOLVE EVPORTFOL USING NLP MAXIMIZING OBJ ;
64
65 SCALAR VAR THE VARIANCE ;
66 VAR = SUM(STOCK, SUM(STOCKS,
67 INVEST.L(STOCK)* COVAR(STOCK,STOCKS) * INVEST.L(STOCKS))) ;
68 DISPLAY VAR ;

```

```

69
70 SET RAPS RISK AVERSION PARAMETERS /R0*R25/
71
72 PARAMETER RISKAVR(RAPS) RISK AVERSION COEFICIENTS
73 /R0 0.00000, R1 0.00025, R2 0.00050, R3 0.00075,
74 R4 0.00100, R5 0.00150, R6 0.00200, R7 0.00300,
75 R8 0.00500, R9 0.01000, R10 0.01100, R11 0.01250,
76 R12 0.01500, R13 0.02500, R14 0.05000, R15 0.10000,
77 R16 0.30000, R17 0.50000, R18 1.00000, R19 2.50000,
78 R20 5.00000, R21 10.0000, R22 15. , R23 20.
79 R24 40. , R25 80./
80
81 PARAMETER OUTPUT(*,RAPS) RESULTS FROM MODEL RUNS WITH VARYING RAP
82
83 OPTION SOLPRINT = OFF;
84
85 LOOP (RAPS,RAP=RISKAVR(RAPS);
86 SOLVE EVPORTFOL USING NLP MAXIMIZING OBJ ;
87 VAR = SUM(STOCK, SUM(STOCKS,
88 INVEST.L(STOCK)* COVAR(STOCK,STOCKS) * INVEST.L(STOCKS))) ;
89 OUTPUT("OBJ",RAPS)=OBJ.L;
90 OUTPUT("RAP",RAPS)=RAP;
91 OUTPUT(STOCKS,RAPS)=INVEST.L(STOCKS);
92 OUTPUT("MEAN",RAPS)=SUM(STOCKS, MEAN(STOCKS)*INVEST.L(STOCKS));
93 OUTPUT("VAR",RAPS) = VAR;
94 OUTPUT("STD",RAPS)=SQRT(VAR);
95 OUTPUT("SHADPRICE",RAPS)=INVESTAV.M;
96 OUTPUT("IDLE",RAPS)=FUNDS-INVESTAV.L
97 );
98 DISPLAY OUTPUT;

```

---

**Table 14.4. E-V Example Solutions for Alternative Risk Aversion Parameters**

RAP	0	0.00025	0.0005	0.00075	0.001
BUYSTOCK2			1.263	5.324	7.355
BUYSTOCK3	17.857	17.857	16.504	12.152	9.977
OBJ	148.214	143.287	138.444	135.688	134.245
MEAN	148.214	148.214	146.581	141.331	138.705
VAR	19709.821	19709.821	16274.764	7523.441	4460.478
STD	140.392	140.392	127.573	86.738	66.787
SHADPRICE	0.296	0.277	0.261	0.260	0.260
RAP	0.0015	0.002	0.003	0.005	0.010
BUYSTOCK2	9.386	10.401	11.416	12.229	12.838
BUYSTOCK3	7.801	6.713	5.625	4.755	4.102
OBJ	132.671	131.753	130.575	129.005	125.999
MEAN	136.080	134.767	133.454	132.404	131.617
VAR	2272.647	1506.907	959.949	679.907	561.764
STD	47.672	38.819	30.983	26.075	23.702
SHADPRICE	0.259	0.257	0.255	0.251	0.241
RAP	0.011	0.012	0.015	0.025	0.050
BUYSTOCK1			1.273	4.372	4.405
BUYSTOCK2	12.893	12.960	12.420	11.070	8.188
BUYSTOCK3	4.043	3.972	3.550	2.561	1.753
BUYSTOCK4					4.168
OBJ	125.441	124.614	123.380	120.375	116.805
MEAN	131.545	131.459	129.839	125.939	121.656
VAR	554.929	547.587	430.560	222.576	97.026
STD	23.557	23.401	20.750	14.919	9.850
SHADPRICE	0.239	0.236	0.234	0.230	0.224
RAP	0.100	0.300	0.500	1.000	2.500
BUYSTOCK1	4.105	3.905	3.865	3.835	1.777
BUYSTOCK2	6.488	5.354	5.128	4.958	2.289
BUYSTOCK3	1.340	1.064	1.009	0.968	0.446
BUYSTOCK4	6.829	8.602	8.957	9.223	4.296
OBJ	113.118	102.254	92.010	66.674	27.185
MEAN	119.327	117.774	117.463	117.230	54.370
VAR	62.086	51.734	50.905	50.556	10.874
STD	7.879	7.193	7.135	7.110	3.298
SHADPRICE	0.214	0.173	0.133	0.032	0
IDLE FUNDS					268.044

Notes: RAP is the risk aversion parameter (M) value  
 Stock<sub>i</sub> gives the amount invested in stock<sub>i</sub>  
 Obj gives the objective function value  
 Mean gives expected income  
 Var gives the variance of income  
 STD gives the standard deviation of income  
 Shadprice gives the shadow price on the capital available constraint

**Table 14.5. Deviations from the Mean for Portfolio Example**

	Stock1	Stock2	Stock3	Stock4
Event1	2.3	-1.6	-0.3	-0.8
Event2	3.3	-3.6	7.7	0.2
Event3	-0.7	0.4	5.7	0.2
Event4	0.3	1.4	-10.3	1.2
Event5	1.3	-0.6	4.7	0.2
Event6	-1.7	2.4	2.7	-0.8
Event7	-2.7	4.4	-10.3	0.2
Event8	0.3	-3.6	9.7	0.2
Event9	-0.7	-0.6	3.7	-0.8
Event10	-1.7	1.4	-13.3	0.2

**Table 14.6. Example MOTAD Model Formulation**

---

Max	4.70 $X_1$	% 7.60 $X_2$	% 8.30 $X_3$	% 5.80 $X_4$		& ( F
s.t.	22 $X_1$	% 30 $X_2$	% 28 $X_3$	% 26 $X_4$		# 500
	%2.300 $X_1$	& 1.600 $X_2$	& 0.300 $X_3$	& 0.800 $X_4$	% $d_1^k$	\$ 0
	%3.300 $X_1$	& 3.600 $X_2$	%7.700 $X_3$	%0.200 $X_4$	% $d_2^k$	\$ 0
	& 0.700 $X_1$	%0.400 $X_2$	%5.700 $X_3$	%0.200 $X_4$	% $d_3^k$	\$ 0
	%0.300 $X_1$	%1.400 $X_2$	& 10.300 $X_3$	%1.200 $X_4$	% $d_4^k$	\$ 0
	%1.300 $X_1$	& 0.600 $X_2$	%4.700 $X_3$	%0.200 $X_4$	% $d_5^k$	\$ 0
	& 1.700 $X_1$	%2.400 $X_2$	%2.700 $X_3$	& 0.800 $X_4$	% $d_6^k$	\$ 0
	& 2.700 $X_1$	%4.400 $X_2$	& 10.300 $X_3$	%0.200 $X_4$	% $d_7^k$	\$ 0
	%0.300 $X_1$	& 3.600 $X_2$	%9.700 $X_3$	%0.200 $X_4$	% $d_8^k$	\$ 0
	& 0.700 $X_1$	& 0.600 $X_2$	%3.700 $X_3$	& 0.800 $X_4$	% $d_9^k$	\$ 0
	& 1.700 $X_1$	%1.400 $X_2$	& 13.300 $X_3$	%0.200 $X_4$	% $d_{10}^k$	\$ 0
					' $d_k^k$ & TND	' 0
					) TND & F	' 0

---

**Table 14.7. MOTAD Example Solutions for Alternative Risk Aversion Parameters**

RAP		0.050	0.100	0.110	0.120
BUYSTOCK2					11.603
BUYSTOCK3	17.857	17.857	17.857	17.857	5.425
OBJ	148.214	140.146	132.078	130.464	129.390
MEAN	148.214	148.214	148.214	148.214	133.213
MAD	122.143	122.143	122.143	122.143	24.111
STDAPPROX	161.367	161.367	161.367	161.367	31.854
VAR	19709.821	19709.821	19709.821	19709.821	883.113
STD	140.392	140.392	140.392	140.392	29.717
SHADPRICE	0.296	0.280	0.264	0.261	0.259
RAP	0.130	0.150	0.260	0.400	0.500
BUYSTOCK1					2.663
BUYSTOCK2	11.603	11.603	11.916	12.379	10.985
BUYSTOCK3	5.425	5.425	5.090	4.594	3.995
OBJ	129.072	128.435	125.179	121.204	118.606
MEAN	133.213	133.213	132.809	132.210	129.161
MAD	24.111	24.111	22.212	20.827	15.979
STDAPPROX	31.854	31.854	29.345	27.515	21.110
VAR	883.113	883.113	771.228	643.507	455.983
STD	29.717	29.717	27.771	25.367	21.354
SHADPRICE	0.258	0.257	0.250	0.242	0.237
RAP	0.750	1.000	1.250	1.500	1.750
BUYSTOCK1	5.145	7.119	2.817	2.817	2.817
BUYSTOCK2	10.409	9.879	5.617	5.617	5.617
BUYSTOCK3	2.661	1.564	1.824	1.824	1.824
BUYSTOCK4		0.123	8.402	8.402	8.402
OBJ	114.168	111.009	108.372	106.086	103.801
MEAN	125.384	122.240	119.799	119.799	119.799
MAD	11.320	8.501	6.920	6.920	6.920
STDAPPROX	14.955	11.231	9.142	9.142	9.142
VAR	211.996	121.386	83.886	83.886	83.886
STD	14.560	11.018	9.159	9.159	9.159
SHADPRICE	0.228	0.222	0.217	0.212	0.208
RAP	2.000	2.500	5.000	10.000	12.500
BUYSTOCK1	2.817	2.817	2.858	2.858	2.858
BUYSTOCK2	5.617	5.617	4.178	4.178	4.178
BUYSTOCK3	1.824	1.824	1.242	1.242	1.242
BUYSTOCK4	8.402	8.402	10.654	10.654	10.654
OBJ	101.515	96.944	76.540	35.790	15.415
MEAN	119.799	119.799	117.289	117.289	117.289
MAD	6.920	6.920	6.169	6.169	6.169
STDAPPROX	9.142	9.142	8.150	8.150	8.150
VAR	83.886	83.886	57.695	57.695	57.695
STD	9.159	9.159	7.596	7.596	7.596
SHADPRICE	0.203	0.194	0.153	0.072	0.031

Note: The abbreviations are the same as in Table 14.4 with the addition of MAD which gives the mean absolute deviation and STDAPPROX which gives the standard deviation approximation.

**Table 14.8. Example Formulation of Safety First Problem**

Max	4.70	X <sub>1</sub>	%	7.60	X <sub>2</sub>	%	8.30	X <sub>3</sub>	%	5.80	X <sub>4</sub>	
s.t.	22	X <sub>1</sub>	%	30	X <sub>2</sub>	%	28	X <sub>3</sub>	%	26	X <sub>4</sub>	# 500
	7	X <sub>1</sub>	%	6	X <sub>2</sub>	%	8	X <sub>3</sub>	%	5	X <sub>4</sub>	\$ S
	8	X <sub>1</sub>	%	4	X <sub>2</sub>	%	16	X <sub>3</sub>	%	6	X <sub>4</sub>	\$ S
	4	X <sub>1</sub>	%	8	X <sub>2</sub>	%	14	X <sub>3</sub>	%	6	X <sub>4</sub>	\$ S
	5	X <sub>1</sub>	%	9	X <sub>2</sub>	&	2	X <sub>3</sub>	%	7	X <sub>4</sub>	\$ S
	6	X <sub>1</sub>	%	7	X <sub>2</sub>	%	13	X <sub>3</sub>	%	6	X <sub>4</sub>	\$ S
	3	X <sub>1</sub>	%	10	X <sub>2</sub>	%	11	X <sub>3</sub>	%	5	X <sub>4</sub>	\$ S
	2	X <sub>1</sub>	%	12	X <sub>2</sub>	&	2	X <sub>3</sub>	%	6	X <sub>4</sub>	\$ S
	5	X <sub>1</sub>	%	4	X <sub>2</sub>	%	18	X <sub>3</sub>	%	6	X <sub>4</sub>	\$ S
	4	X <sub>1</sub>	%	7	X <sub>2</sub>	%	12	X <sub>3</sub>	%	5	X <sub>4</sub>	\$ S
	3	X <sub>1</sub>	%	9	X <sub>2</sub>	&	5	X <sub>3</sub>	%	6	X <sub>4</sub>	\$ S

**Table 14.9. Safety First Example Solutions for Alternative Safety Levels**

RUIN	-100.000	-50.000	0.0	25.000	50.000
BUYSTOCK2	0.0	2.736	6.219	7.960	9.701
BUYSTOCK3	17.857	14.925	11.194	9.328	7.463
OBJ	148.214	144.677	140.174	137.923	135.672
MEAN	148.214	144.677	140.174	137.923	135.672
VAR	19709.821	12695.542	6066.388	3717.016	2011.116
STD	140.392	112.674	77.887	60.967	44.845
SHADPRICE	0.296	0.280	0.280	0.280	0.280

Note: The abbreviations are the same as in the previous example solutions with RUIN giving the safety level.

**Table 14.10. Example Formulation of Target MOTAD Problem**

---

Max	4.70 $X_1$	%	7.60 $X_2$	%	8.30 $X_3$	%	5.80 $X_4$			
s.t.	22 $X_1$	%	30 $X_2$	%	28 $X_3$	%	26 $X_4$		#	500
	7 $X_1$	%	6 $X_2$	%	8 $X_3$	%	5 $X_4$	%	Dev <sub>1</sub>	\$ T
	8 $X_1$	%	4 $X_2$	%	16 $X_3$	%	6 $X_4$	%	Dev <sub>2</sub>	\$ T
	4 $X_1$	%	8 $X_2$	%	14 $X_3$	%	6 $X_4$	%	Dev <sub>3</sub>	\$ T
	5 $X_1$	%	9 $X_2$	&	2 $X_3$	%	7 $X_4$	%	Dev <sub>4</sub>	\$ T
	6 $X_1$	%	7 $X_2$	%	13 $X_3$	%	6 $X_4$	%	Dev <sub>5</sub>	\$ T
	3 $X_1$	%	10 $X_2$	%	11 $X_3$	%	5 $X_4$	%	Dev <sub>6</sub>	\$ T
	2 $X_1$	%	12 $X_2$	&	2 $X_3$	%	6 $X_4$	%	Dev <sub>7</sub>	\$ T
	5 $X_1$	%	4 $X_2$	%	18 $X_3$	%	6 $X_4$	%	Dev <sub>8</sub>	\$ T
	4 $X_1$	%	7 $X_2$	%	12 $X_3$	%	5 $X_4$	%	Dev <sub>9</sub>	\$ T
	3 $X_1$	%	9 $X_2$	&	5 $X_3$	%	6 $X_4$	%	Dev <sub>10</sub>	\$ T
									Dev <sub>k</sub>	# 8

---



**Table 14.11. Target MOTAD Example Solutions for Alternative Deviation Limits**

---

TARGETDEV	120.000	60.000	24.000	12.000	10.800
BUYSTOCK2	0.0	0.0	7.081	10.193	10.516
BUYSTOCK3	17.857	17.857	10.270	6.936	6.590
OBJ	148.214	148.214	139.059	135.037	134.618
MEAN	148.214	148.214	139.059	135.037	134.618
VAR	19709.821	19709.821	4822.705	1646.270	1433.820
STD	140.392	140.392	69.446	40.574	37.866
SHADPRICE	0.296	0.296	0.286	0.295	0.295
TARGETDEV	8.400	7.200	3.600		
BUYSTOCK1	0.0	0.0	3.459		
BUYSTOCK2	11.259	11.782	11.405		
BUYSTOCK3	5.794	5.234	2.919		
OBJ	133.659	132.982	127.168		
MEAN	133.659	132.982	127.168		
VAR	1030.649	816.629	277.270		
STD	32.104	28.577	16.651		
SHADPRICE	0.298	0.298	0.815		

---

Note: The abbreviations are again the same with TARGETDEV giving the 8 value.

**Table 14.12. Example Formulation of DEMP Problem**

---

Max		$(W_k)^{\text{power}}$													
s.t.			22	X <sub>1</sub>	%	30	X <sub>2</sub>	%	28	X <sub>3</sub>	%	26	X <sub>4</sub>	#	500
	W <sub>1</sub>	&	7	X <sub>1</sub>	&	6	X <sub>2</sub>	&	8	X <sub>3</sub>	&	5	X <sub>4</sub>	'	100
	W <sub>2</sub>	&	8	X <sub>1</sub>	&	4	X <sub>2</sub>	&	16	X <sub>3</sub>	&	6	X <sub>4</sub>	'	100
	W <sub>3</sub>	&	4	X <sub>1</sub>	&	8	X <sub>2</sub>	&	14	X <sub>3</sub>	&	6	X <sub>4</sub>	'	100
	W <sub>4</sub>	&	5	X <sub>1</sub>	&	9	X <sub>2</sub>	%	2	X <sub>3</sub>	&	7	X <sub>4</sub>	'	100
	W <sub>5</sub>	&	6	X <sub>1</sub>	&	7	X <sub>2</sub>	&	13	X <sub>3</sub>	&	6	X <sub>4</sub>	'	100
	W <sub>6</sub>	&	3	X <sub>1</sub>	&	10	X <sub>2</sub>	&	11	X <sub>3</sub>	&	5	X <sub>4</sub>	'	100
	W <sub>7</sub>	&	2	X <sub>1</sub>	&	12	X <sub>2</sub>	%	2	X <sub>3</sub>	&	6	X <sub>4</sub>	'	100
	W <sub>8</sub>	&	5	X <sub>1</sub>	&	4	X <sub>2</sub>	&	18	X <sub>3</sub>	&	6	X <sub>4</sub>	'	100
	W <sub>9</sub>	&	4	X <sub>1</sub>	&	7	X <sub>2</sub>	&	12	X <sub>3</sub>	&	5	X <sub>4</sub>	'	100
	W <sub>10</sub>	&	3	X <sub>1</sub>	&	9	X <sub>2</sub>	%	5	X <sub>3</sub>	&	6	X <sub>4</sub>	'	100

---

**Table 14.13. DEMP Example Solutions for Alternative Utility Function Exponents**

POWER	0.950	0.900	0.750	0.500	0.400
BUYSTOCK2			4.560	8.563	9.344
BUYSTOCK3	17.857	17.857	12.972	8.683	7.846
OBJ	186.473	140.169	60.363	15.282	8.848
MEAN	248.214	248.214	242.319	237.144	236.134
VAR	19709.821	19709.821	8903.295	3054.034	2309.233
STD	140.392	140.392	94.357	55.263	48.054
SHADPRICE	0.287	0.277	0.269	0.266	0.265
POWER	0.300	0.200	0.100	0.050	0.030
BUYSTOCK2	9.919	10.358	10.705	10.852	10.907
BUYSTOCK3	7.230	6.759	6.388	6.230	6.171
OBJ	5.127	2.972	1.724	1.313	1.177
MEAN	235.390	234.822	234.374	234.184	234.113
VAR	1843.171	1534.736	1320.345	1236.951	1207.076
STD	42.932	39.176	36.337	35.170	34.743
SHADPRICE	0.264	0.264	0.263	0.263	0.263
POWER	0.020	0.010	0.001	0.0001	
BUYSTOCK2	10.934	10.960	10.960	10.960	
BUYSTOCK3	6.143	6.115	6.115	6.115	
OBJ	1.115	1.056	1.005	1.001	
MEAN	234.079	234.045	234.045	234.045	
VAR	1192.805	1178.961	1178.961	1178.961	
STD	34.537	34.336	34.336	34.336	
SHADPRICE	0.263	0.263	0.263	0	

Note: The abbreviations are again the same with POWER giving the exponent used.

**Table 14.14. Chance Constrained Example Data**

Event	Small Lathe	Large Lathe	Carver
1	140	90	120
2	120	94	132
3	133	88	110
4	154	97	118
5	133	87	133
6	142	86	107
7	155	90	120
8	140	94	114
9	142	89	123
10	141	85	123
Mean	140	90	120
Standard Error	9.63	3.69	8.00

**Table 14.15. Chance Constrained Example Solutions for Alternative Alpha Levels**

Z-	0.00	1.280	1.654	2.330
PROFIT	10417.291	9884.611	9728.969	9447.647
SMLLATHE	140.000	127.669	124.067	117.554
LRGLATHE	90.000	85.280	83.900	81.407
CARVER	120.000	109.760	106.768	101.360
LABOR	125.000	125.000	125.000	125.000
FUNCTNORM	62.233	78.102	82.739	91.120
FANCYNORM	73.020	51.495	45.205	33.837
FANCYMLRG	5.180	6.788	7.258	8.108

Note: Z is the risk aversion parameter.

**Table 14.16. Feed Nutrients by State of Nature for Wicks Guise Example**

---

Nutrient	State	CORN	SOYBEANS	WHEAT
ENERGY	S1	1.15	0.26	1.05
ENERGY	S2	1.10	0.31	0.95
ENERGY	S3	1.25	0.23	1.08
ENERGY	S4	1.18	0.28	1.12
PROTEIN	S1	0.23	1.12	0.51
PROTEIN	S2	0.17	1.08	0.59
PROTEIN	S3	0.25	1.01	0.46
PROTEIN	S4	0.15	0.99	0.56

---

**Table 14.17. Wicks Guise Example**

	Corn	Soybeans	Wheat	EnDev	EnMAD	EnF	PrDev	PrMAD	PrF
Objective	0.03	0.06	0.04						
Volume	1	1	1						1
Energy	1.17	0.27	1.05			& N			\$ 0.8
Protein	0.20	1.05	0.53						& N \$ 0.5
Energys1	& 0.02	& 0.01	% 0.00	& $d_{e1}^{\%}$	$d_{e1}^{\&}$				' 0
Energys2	& 0.07	% 0.04	& 0.10	& $d_{e2}^{\%}$	$d_{e2}^{\&}$				' 0
Energys3	% 0.08	& 0.04	% 0.03	& $d_{e3}^{\%}$	$d_{e3}^{\&}$				' 0
Energys4	% 0.01	% 0.01	% 0.07	& $d_{e4}^{\%}$	$d_{e4}^{\&}$				' 0
EnergyMAD				$(d_{ek}^{\%} d_{ek}^{\&})/4$	& 1				' 0
EnergyF					& )	% 1			' 0
Proteins1	& 0.02	& 0.01	% 0.00				& $d_{p1}^{\%}$	$d_{p1}^{\&}$	' 0
Proteins2	& 0.07	% 0.04	& 0.10				& $d_{p2}^{\%}$	$d_{p2}^{\&}$	' 0
Proteins3	% 0.08	& 0.04	% 0.03				& $d_{p3}^{\%}$	$d_{p3}^{\&}$	' 0
Proteins4	% 0.01	% 0.01	% 0.07				& $d_{p4}^{\%}$	$d_{p4}^{\&}$	' 0
ProteinMAD							$(d_{pk}^{\%} d_{pk}^{\&})/4$	& 1	' 0
ProteinF								& )	% 1 ' 0

Note: EnDev is the energy deviation  
 EnMAD is the energy mean absolute deviation  
 EnF is the energy standard deviation approximations  
 PrDev is the protein deviation  
 PrMAD is the protein mean absolute deviation  
 PrF is the protein standard deviation approximation

**Table 14.18. Results From Example Wicks Guise Model Runs With Varying RAP**

RAP		0.250	0.500	0.750	1.000
CORN	0.091	0.046	0.211	0.230	0.221
SOYBEANS			0.105	0.129	0.137
WHEAT	0.909	0.954	0.684	0.641	0.642
OBJ	0.039	0.040	0.040	0.040	0.041
AVGPROTEIN	0.500	0.515	0.515	0.521	0.529
STDPROTEIN	0.054	0.059	0.030	0.028	0.029
AVGENERGY	1.061	1.056	0.993	0.977	0.969
STDENERGY	0.072	0.072	0.061	0.059	0.058
SHADPROT	0.030	0.033	0.036	0.037	0.038
RAP	1.250	1.500	2.000		
CORN	0.211	0.200	0.177		
SOYBEANS	0.146	0.156	0.176		
WHEAT	0.643	0.644	0.647		
OBJ	0.041	0.041	0.042		
AVGPROTEIN	0.536	0.545	0.563		
STDPROTEIN	0.029	0.030	0.031		
AVGENERGY	0.961	0.953	0.934		
STDENERGY	0.057	0.056	0.055		
SHADPROT	0.039	0.040	0.042		

Note: RAP gives the risk aversion parameter used  
 CORN gives the amount of corn used in the solution  
 SOYBEANS gives the amount of soybeans used in the solution  
 WHEAT gives the amount of wheat used in the solution  
 OBJ gives the objective function value  
 AVGPROTEIN gives the average amount of protein in the diet  
 STDPROTEIN gives the standard error of protein in the diet  
 AVGENERGY gives the average amount of energy in the diet  
 STDENERGY gives the standard error of energy in the diet  
 SHADPROT gives the shadow price on the protein requirement constraint

**Table 14.19. Data on Uncertain Parameters in First SPR Example**

Parameter	Value Under	
	State of Nature 1	State of Nature 2
Probability	.6	.4
Corn Yield in bu	100	105
Wheat Yield in bu	40	38
Corn Harvest Rate hours per bu	.010	.015
Wheat Harvest Rate hours per bu	.030	.034
Corn Price per bu	3.25	2.00
Wheat Price per bu	5.00	6.00
Harvest Time hours	122	143

	Grow Corn	Grow Wheat	Income	Harvest Corn	Harvest Wheat	RHS
Objective			1			
Land	1	1				# 100
Corn Yield Balance	-yield <sub>c</sub>			1		# 0
Wheat Yield Balance		-yield <sub>w</sub>			1	# 0
Harvest Hours				+harvtime <sub>c</sub>	+harvtime <sub>w</sub>	# harvavail
Income	-100	-60	-1	+price <sub>c</sub>	+price <sub>w</sub>	= 0



Table 14.21. Formulation of First SPR Example

	State 1				State 2				RHS
	Grow Corn	Grow Wht.	Inc. s1	Harv Corn s1	Harv Wht s1	Inc. s2	Harv Corn s2	Harv Wht s2	
Objective			.6			.4			max
Land	1	1							# 100
Corn s1	-100			1					# 0
Wheat s1		-40			1				# 0
Harvest Hours s1				.010	.030				# 122
Income s1	-100	-60	-1	3.25	5.00				= 0
Corn s2	-105						1		# 0
Wheat s2		-38						1	# 0
Harvest Hours s2						.015	.034		# 143
Income s2	-100	-60				-1	2.00	6.00	= 0

S t a t e 1  
S t a t e 2

**Table 14.22. Solution of First SPR Example**

Equation	Slack	Shadow Price
Objective	16476	
Land	0	24.28
Corn s1	0	-1.95
Wheat s1	0	0.67
Harvest Hours s1	11.75	0
Income s1	0	-0.6
Corn s2	0	-3.00
Wheat s2	0	0.94
Harvest Hours s2	0	98.23
Income s2	0	-0.4

Variable	Solution Value	Marginal Cost
Grow Corn	48.8	0
Grow Wheat	51.2	0
Income S1	18144	0
Harvest Corn s1	4876	0
Harvest Wheat s1	2049	0
Income S2	13972	0
Harvest Corn s2	5120	0
Harvest Wheat s2	1947	0

**Table 14.23. Second SPR Example Formulation (Partial Tableau)**

	Corn	Soy	Whit	Avg	Pos	Neg	Pos	Neg	Pos	Neg	Pos	Neg	Pos	Neg	Pos	Neg		
				Cost	Prot	Eng	Prot	Eng	Prot	Eng	Prot	Eng	Prot	Eng	Prot	Eng		
					Dev	Dev	Dev	Dev	Dev	Dev	Dev	Dev	Dev	Dev	Dev	Dev		
					s1	s1	s1	s1	s1	s1	s1	s1	s2	s2	s2	s2		
					Dev	Dev	Dev	Dev	Dev	Dev	Dev	Dev	Cost	Cost	Cost	Cost		
					s1	s1	s1	s1	s1	s1	s1	s1	Dev	Dev	Dev	Dev		
					s1	s1	s1	s1	s1	s1	s1	s1	Cost	Cost	Cost	Cost		
Objective				1														
Total Feed	1	1	1														1	
Average Cost				1													0	
Protein-s1	0.23	1.12	0.51		-1	1											0.6	
Energy -s1	1.15	0.26	1.05			-1	1										0.9	
Cost-s1	0.03	0.06	0.04		0.50	1.50	1.00	0.10	-1								0	
Cost dev s1				-1					1	-1	1						0	
Protein-s2	0.17	1.08	0.59										-1	1			0.6	
Energy -s2	1.10	0.31	0.95											-1	1		0.9	
Cost-s2	.03	.06	.04										0.50	1.50	1.00	0.10	-1	
Cost dev s2				-1												1	-1	1

**Table 14.24. Second SPR Example Risk Neutral Solution**

	Slack	Shadow Price		Slack	Shadow Price
Objective	0.067		Corn Purchase	0.283	0
Total Feed	0	-0.14	Soybean Purchase	0.362	0
Average Cost	0.00	1.	Wheat Purchase	0.355	0
Protein-s1	0	0.125	Average Cost	0.067	0
Energy -s1	0	0.025	Pos Protein Dev s1	0.052	0
Cost-s1	0	252.66	Neg Protein Dev s1	0.	0.50
Cost dev s1	0	0.00	Pos Energy Dev s1	0.00	0
Protein-s2	0	0.125	Neg Energy Dev s1	0.108	0
Energy -s2	0	0.025	Cost - s1	0.081	0
Cost-s2	0	0.25	Pos Cost Dev - s1	0.014	0
Cost dev s2	0	0	Neg Cost Dev - s1	0.00	0
Protein-s3	0	-.366	Pos Protein Dev s2	0.049	0
Energy -s3	0	0.025	Neg Protein Dev s2	0.000	0.50
Cost-s3	0	0.25	Pos Energy Dev s2	0.	0.275
Cost dev s3	0	0	Neg Energy Dev s2	0.140	0
Protein-s4	0	.08	Cost - s2	0.083	0
Energy -s4	0	.025	Pos Cost Dev - s2	.016	0
Cost-s4	0	0.25	Neg Cost Dev - s2	0.00	0
Cost dev s4	0	0.00	Pos Protein Dev s3	0.	0.491
			Neg Protein Dev s3	0.	0.009
			Pos Energy Dev s3		0.275
			Neg Energy Dev s3	0.080	0
			Cost - s3	0.052	0
			Pos Cost Dev - s3	0.00	0
			Neg Cost Dev - s3	0.014	0
			Pos Protein Dev s4	0.	0.205
			Neg Protein Dev s4	0.	0.295
			Pos Energy Dev s4	0.	0.275
			Neg Energy Dev s4	0.067	0
			Cost - s4	0.051	0
			Pos Cost Dev - s4	0.	0
			Neg Cost Dev - s4	0.016	0

**Table 14.25. SPR Second Example Problem Soution Under Varying Risk Aversion**

RAP	0	0.1	0.2	0.3	0.4	0.500	0.600
Corn	0.283	0.249	0.245	0.244	0.288	0.296	0.297
Soybeans	0.362	0.330	0.327	0.326	0.340	0.342	0.342
Wheat	0.355	0.422	0.428	0.430	0.372	0.363	0.361
Avgcost	0.067	0.067	0.067	0.067	0.071	0.071	0.071
Cost s1	0.081	0.074	0.073	0.073	0.071	0.071	0.071
Cost s2	0.083	0.080	0.080	0.080	0.074	0.073	0.073
Cost s3	0.052	0.066	0.067	0.068	0.071	0.071	0.071
Cost s4	0.051	0.048	0.048	0.048	0.067	0.070	0.071
Std Error	0.015	0.012	0.012	0.012	0.002	0.001	0.001

RAP is the risk aversion parameter.



**Table 14.27. Solution for Third SPR Example**

Variable	Value	Reduced Cost	Variable	Slack	Shadow Price
Average Ending Net Worth	229.748	0	Objective	229.748	
Sell In Period 1	0	-0.162	Starting Stock	0	2.297
Keep From Period 1 to 2	100	0	Avg End Worth	0	1
Sell In Period 2 Under State 1	100	0	Stock Kept pd 1 to 2 s1	0	1.62
Keep From Period 2 to 3 Under State 1	0	-0.021	Stock Kept pd 1 to 2 s1	0	0.677
Sell In Period 2 Under State 2	0	-0.027	Stock Kept pd 2 to 3 s1-s1	0	0.916
Keep From Period 2 to 3 Under State 2	100	0	Ending Worth s1-s1	0	-0.42
Sell In Period 3 Under State 1 -- State A	0	0	Stock Kept pd 2 to 3 s1-s2	0	0.683
Ending Worth Under State 1 -- State A	233.2	0	Ending Worth s1-s2	0	-0.28
Sell In Period 3 Under State 1 -- State B	0	0	Stock Kept pd 2 to 3 s2-s1	0	0.458
Ending Worth Under State 1 -- State B	228.8	0	Ending Worth s2-s1	0	-0.21
Sell In Period 3 Under State 2 -- State A	100	0	Stock Kept pd 2 to 3 s2-s2	0	0.22
Ending Worth Under State 2 -- State A	218	0	Ending Worth s2-s2	0	-0.09
Sell In Period 3 Under State 2 -- State B	100	0			
Ending Worth Under State 2 -- State B	244	0			

### 14.1. E-V Model Efficient Frontier

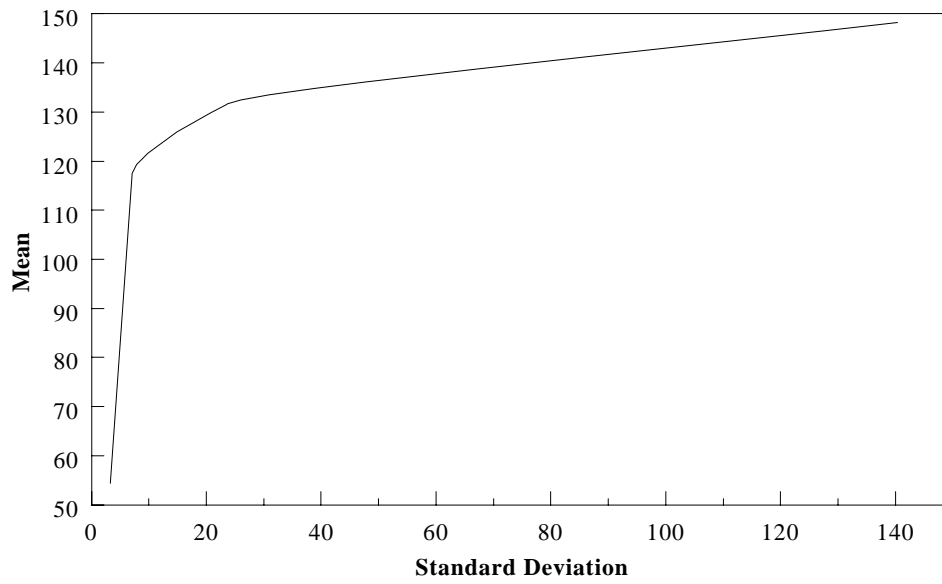
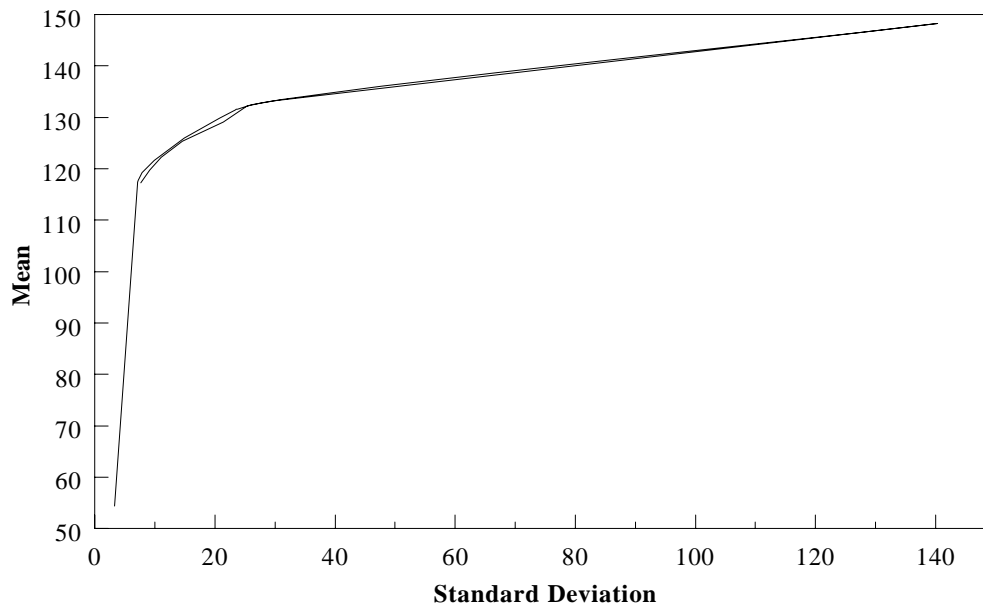
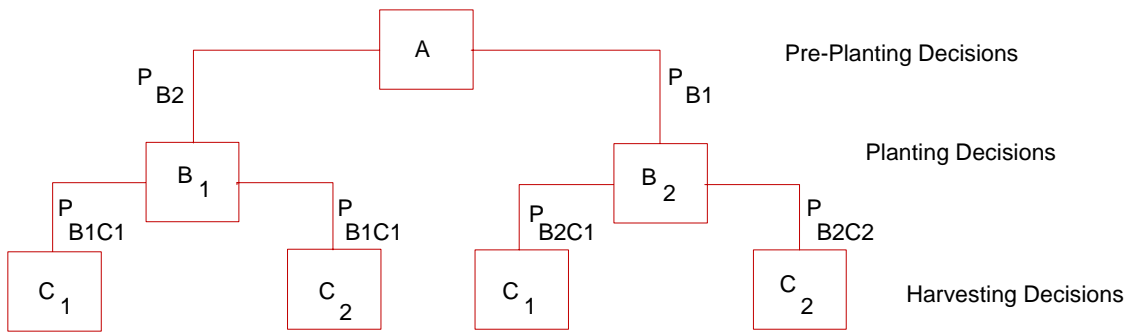


Figure 14.2. E-V and MOTAD Efficient Frontiers







**Figure 14.3. Decision Tree for Sequential Programming Example**

## CHAPTER XV: APPLIED INTEGER PROGRAMMING

LP assumes continuity of the solution region. LP decision variables can equal whole numbers or any other real number (3 or 4 as well as 3.49876). However, fractional solutions are not always acceptable. Particular items may only make sense when purchased in whole units (e.g., tractors, or airplanes). Integer programming (IP) requires a subset of the decision variables to take on integer values (i.e., 0, 1, 2, etc.). IP also permits modeling of fixed costs, logical conditions, discrete levels of resources and nonlinear functions.

IP problems usually involve optimization of a linear objective function subject to linear constraints, nonnegativity conditions and integer value conditions. The integer valued variables are called integer variables. Problems containing integer variables fall into several classes. A problem in which all variables are integer is a pure integer IP problem. A problem with some integer and some continuous variables, is a mixed-integer IP problem. A problem in which the integer variables are restricted to equal either zero or one is called a zero-one IP problem. There are pure zero-one IP problems where all variables are zero-one and mixed zero-one IP problems containing both zero-one and continuous variables. The most general formulation of the IP problem is:

$$\begin{array}{llllll}
 \text{Max} & C_1W & \% & C_2X & \% & C_3Y \\
 \text{s.t.} & A_1W & \% & A_2X & \% & A_3Y & \# & b \\
 & W & & & & & \$ & 0 \\
 & & & X & & & \$ & 0 \text{ and integer} \\
 & & & & & Y & ' & 0 \text{ or } 1
 \end{array}$$

where the W's represent continuous variables; the X's integer variables, and the Y's zero-one variables.

Our coverage of integer programming is divided into two chapters. This chapter covers basic integer programming problem formulation techniques, and a few characteristics relative to the solution and interpretation of integer programming problems. The next chapter goes into a set of example problems.

### 15.1 Why Integer Programming

The most fundamental question regarding the use of integer programming (IP) is why use it. Obviously, IP allows one to depict discontinuous decision variables, such as those representing acquisition of indivisible items such as machines, hired labor or animals. In addition, IP also permits modeling of fixed costs, logical conditions, and discrete levels of resources as will be discussed here.

#### 15.1.1 Fixed Cost

Production processes often involve fixed costs. For example, when manufacturing multiple products,

fixed costs may arise when shifting production between products (i.e., milk plant operators must incur cleaning costs when switching from chocolate to white milk). Fixed costs can be modeled using the following mixed integer formulation strategy:

- Let:
- X denote the continuous number of units of a good produced;
  - Y denote a zero-one variable indicating whether or not fixed costs are incurred;
  - C denote the per unit revenue from producing X;
  - F denote the fixed cost incurred when producing a nonzero quantity of regardless of how many units are produced; and
  - M denote a large number.

The formulation below depicts this problem:

$$\begin{aligned}
 & \text{Max } CX - FY \\
 & \text{s.t. } X - MY \leq 0 \\
 & \quad X \leq M \\
 & \quad Y \in \{0, 1\}
 \end{aligned}$$

Here, when  $X = 0$ , the constraint relating  $X$  and  $Y$  allows  $Y$  to be 0 or 1. Given  $F > 0$  then the objective function would cause  $Y$  to equal 0. However, when  $0 < X \leq M$ , then  $Y$  must equal 1. Thus, any non-zero production level for  $X$  causes the fixed cost ( $F$ ) to be incurred. The parameter  $M$  is an upper bound on the

production of X (a capacity limit).

The fixed cost of equipment investment may be modeled similarly. Suppose one is modeling the possible acquisition of several different-sized machines, all capable of doing the same task. Further, suppose that per unit profits are independent of the machine used, that production is disaggregated by month, and that each machine's monthly capacity is known. This machinery acquisition and usage decision problem can be formulated as:

$$\begin{aligned}
 \text{Max} \quad & \sum_m C_m X_m \quad \& \quad \sum_k F_k Y_k \\
 \text{s.t.} \quad & \sum_m X_m \quad \& \quad \sum_k \text{Cap}_{km} Y_k \quad \# \quad 0 \quad \text{for all } m \\
 & X_m \quad \$ \quad 0, \quad Y_k \quad \in \quad \{0, 1\} \quad \text{for all } k \text{ and } m,
 \end{aligned}$$

where  $m$  denotes month,  $k$  denotes machine,  $C_m$  is the profit obtained from production in month  $m$ ;  $X_m$  is the quantity produced in month  $m$ ;  $F_k$  is the annualized fixed cost of the  $k^{\text{th}}$  machine;  $Y_k$  is a zero-one variable indicating whether or not the  $k^{\text{th}}$  machine is purchased; and  $\text{Cap}_{km}$  is the capacity of the  $k^{\text{th}}$  machine in the  $m^{\text{th}}$  month.

The overall formulation maximizes annual operating profits minus fixed costs subject to constraints that permit production only when machinery has been purchased. Purchase of several machinery items with different capacity characteristics is allowed. This formulation permits  $X_m$  to be non-zero only when at least one  $Y_k$  is non-zero. Again, machinery must be purchased with the fixed cost incurred before it is used. Once purchased any machine allows production up to its capacity in each of the 12 months. This formulation illustrates a link between production and machinery purchase (equivalently purchase and use of a piece of capital equipment) through the capacity constraint. One must be careful to properly specify the fixed costs so that they represent the portion of cost incurred during the time-frame of the model.

### 15.1.2 Logical Conditions

IP also allows one to depict logical conditions. Some examples are:

- a) Conditional Use - A warehouse can be used only if constructed.
- b) Complementary Products - If any of product A is produced, then a minimum quantity of product B must be produced.
- c) Complementary Investment - If a particular class of equipment is purchased then only complementary equipment can be acquired.
- d) Sequencing - Operation A must be entirely complete before operation B starts.

All of these conditions can be imposed using a zero-one indicator variable. An indicator variable tells whether a sum is zero or nonzero. The indicator variable takes on a value of one if the sum is nonzero and zero otherwise. An indicator variable is imposed using a constraint like the following:

$$\sum_i X_i \leq MY \quad \& \quad MY \leq M$$

where  $M$  is a large positive number,  $X_i$  depicts a group of continuous variables, and  $Y$  is an indicator variable restricted to be either zero or one. The indicator variable  $Y$  indicates whether or not any of the  $X$ 's are non-zero with  $Y=1$  if so, zero otherwise. Note this formulation requires that  $M$  must be as large as any reasonable value for the sum of the  $X$ 's.

Indicator variables may be used in many ways. For example, consider a problem involving two mutually exclusive products,  $X$  and  $Z$ . Such a problem may be formulated using the constraints

$$\begin{aligned}
X & \leq MY_1 & & \leq 0 \\
Z & & \leq MY_2 & \leq 0 \\
Y_1 + Y_2 & & & \leq 1 \\
X, Z & & & \leq 0 \\
Y_1, Y_2 & & & \in \{0, 1\}
\end{aligned}$$

Here,  $Y_1$  indicates whether or not  $X$  is produced, while  $Y_2$  indicates whether or not  $Z$  is produced. The third constraint,  $Y_1 + Y_2 \leq 1$ , in conjunction with the zero-one restriction on  $Y_1$  and  $Y_2$ , imposes mutual exclusivity. Thus, when  $Y_1 = 1$  then  $X$  can be produced but  $Z$  cannot. Similarly, when  $Y_2 = 1$  then  $X$  must be zero while  $0 \leq Z \leq M$ . Consequently, either  $X$  or  $Z$  can be produced, but not both.

#### 15.1.2.1 Either-or-Active Constraints

Many types of logical conditions may be modeled using indicator variables and mutual exclusivity. Suppose only one of two constraints is to be active, i.e.,

$$\begin{aligned}
& \text{either } A_1X \leq b_1 \\
& \text{or } A_2X \leq b_2
\end{aligned}$$

Formulation of this situation may be accomplished utilizing the indicator variable  $Y$  as follows

$$\begin{aligned}
A_1X & \leq MY & \leq b_1 \\
A_2X & \leq M(1 - Y) & \leq b_2 \\
X & \geq 0, \quad Y \in \{0, 1\}
\end{aligned}$$

This is rewritten as

$$\begin{aligned}
A_1X & \leq MY & \leq b_1 \\
A_2X & \leq MY & \leq b_2 - M \\
X & \geq 0, \quad Y \in \{0, 1\}
\end{aligned}$$

Here  $M$  is a large positive number and the value of  $Y$  indicates which constraint is active. When  $Y = 1$  the second constraint is active while the first constraint is removing it from active consideration. Conversely, when  $Y = 0$  the first constraint is active.

#### 15.1.2.2 An Aside: Mutual Exclusivity

The above formulation contains a common trick for imposing mutual exclusivity. The formulation could have been written as:

$$\begin{array}{r}
A_1 X & + & M Y_1 & \leq & b_1 \\
A_2 X & & & + & M Y_2 & \leq & b_2 \\
& & Y_1 & + & Y_2 & = & 1 \\
X & \geq & 0, & Y_1, & Y_2 & = & 0 \text{ or } 1.
\end{array}$$

However, one can solve for  $Y_2$  in the third constraint yielding  $Y_2 = 1 - Y_1$ . In turn, substituting in the first two equations gives

$$\begin{array}{r}
A_1 X + M Y_1 \leq b_1 \\
A_2 X + M(1 - Y_1) \leq b_2
\end{array}$$

which is the compact formulation above. However, Williams (1978b) indicates that the more extensive formulation will solve faster.

### 15.1.2.3 Multiple Active Constraints

The formulation restricting the number of active constraints may be generalized to logical conditions where  $P$  out of  $K$  constraints are active ( $P < K$ ). This is represented by

$$\begin{array}{r}
A_1 X + M Y_1 \leq b_1 \\
A_2 X + M Y_2 \leq b_2 \\
\vdots \\
\vdots \\
A_K X + M Y_K \leq b_K \\
\sum_i Y_i = K - P \\
X \geq 0, Y_i = 0 \text{ or } 1 \text{ for all } i
\end{array}$$

Here,  $Y_i$  identifies whether constraint  $i$  is active ( $Y_i = 0$ ) or not ( $Y_i = 1$ ). The last constraint requires exactly  $K - P$  of the  $K$  constraints to be non-active, thus  $P$  constraints are active.

### 15.1.2.4 Conditional Restrictions

Logical conditions and indicator variables are useful in imposing conditional restrictions. For example, nonzero values in one group of variables ( $X$ ) might imply nonzeros for another group of variables ( $Y$ ). This may be formulated as

$$\begin{array}{r}
\sum_i X_i + M Z \leq 0 \\
\sum_k Y_k + R Z \leq 0 \\
X_i, Y_k \geq 0, Z = 0 \text{ or } 1
\end{array}$$

Here  $X_i$  are the elements of the first group;  $Z$  is an indicator variable indicating whether any  $X_i$  has been purchased;  $Y_k$  are the elements of the second group; and  $M$  is a large number.  $Z$  can be zero only if all the



X's are 0 and must be one otherwise. The sum of the Y's must be greater than R if the indicator variable Z is one.

### 15.1.3 Discrete Levels of Resources

Situations may arise where variables are constrained by discrete resource conditions. For example, suppose a farm has three fields. Farmers usually plant each field to a single crop. Thus, a situation might require crops to be grown in acreages consistent with entire fields being planted to a single crop. This restriction can be imposed using indicator variables. Assume that there are 3 fields of sizes  $F_1$ ,  $F_2$ , and  $F_3$ , each of which must be totally allocated to either crop 1 ( $X_1$ ) or crop 2 ( $X_2$ ). Constraints imposing such a condition are

$$\begin{array}{l}
 X_1 \ \& \ F_1 Y_1 \ \& \ F_2 Y_2 \ \& \ F_3 Y_3 \ \ ' \ 0 \\
 X_2 \ \& \ F_1(1 - Y_1) \ \& \ F_2(1 - Y_2) \ \& \ F_3(1 - Y_3) \ \ ' \ 0 \\
 \text{or} \\
 X_2 \ \% \ F_1 Y_1 \ \% \ F_2 Y_2 \ \% \ F_3 Y_3 \ \ ' \ F_1 \% F_2 \% F_3 \\
 X_k \ \$ \ 0, \ Y_i \ ' \ 0 \text{ or } 1 \quad \text{for all } k \text{ and } i
 \end{array}$$

The variable  $Y_i$  indicates whether field  $i$  is planted to crop 1 ( $Y_i=1$ ) or crop 2 ( $Y_i=0$ ). The  $X_i$  variables equal the total acreage of crop  $i$  which is planted. For example, when  $Y_1=1$  and  $Y_2, Y_3 = 0$ , then the acreage of crop 1 ( $X_1$ ) will equal  $F_1$  while the acreage of crop 2 ( $X_2$ ) will equal  $F_2 + F_3$ . The discrete variables insure that the fields are assigned in a mutually exclusive fashion.

### 15.1.4 Distinct Variable Values

Situations may require that decision variables exhibit only certain distinct values (i.e., a variable restricted to equal 2, 4, or 12). This can be formulated in two ways. First, if the variable can take on distinct values which exhibit no particular pattern then:

$$\begin{array}{l}
 X \ \& \ V_1 Y_1 \ \& \ V_2 Y_2 \ \& \ V_3 Y_3 \ \ ' \ 0 \\
 Y_1 \ \% \ Y_2 \ \% \ Y_3 \ \ ' \ 1 \\
 X \ \$ \ 0, \ Y \ ' \ 0 \text{ or } 1.
 \end{array}$$

Here, the variable  $X$  can take on either the discrete value of  $V_1$ ,  $V_2$ , or  $V_3$ , where  $V_i$  may be any real number. The second constraint imposes mutual exclusivity between the allowable values.

On the other hand, if the values fall between two limits and are separated by a constant interval, then a different formulation is applicable. The formulation to be used depends on whether zero-one or integer variables are used. When using zero-one variables, a binary expansion is employed. If, for example,  $X$  were restricted to be an integer between 5 and 20 the formulation would be:

$$\begin{array}{l}
 X \ \& \ Y_1 \ \& \ 2Y_2 \ \& \ 4Y_3 \ \& \ 8Y_4 \ \ ' \ 5 \\
 X \ \$ \ 0, \ Y_i \ ' \ 0 \text{ or } 1
 \end{array}$$

Here each  $Y_i$  is a zero-one indicator variable, and  $X$  is a continuous variable, but in the solution,  $X$  will equal an integer value. When all the  $Y$ 's equal zero, then  $X = 5$ . If  $Y_2$  and  $Y_4$  both equal 1 then  $X = 15$ . Through this representation, any integer value of  $X$  between 5 and 20 can occur. In general through the use of  $N$  zero-one variables, any integer value between the right hand side and the right hand side plus  $2^N - 1$  can be represented. Thus, the constraint

$$X \leq a + \sum_{k=1}^N 2^{k-1} Y_k$$

restricts  $X$  to be any integer number between  $a$  and  $a + 2^N - 1$ . This formulation permits one to model general integer values when using a zero-one IP algorithm.

### 15.1.5 Nonlinear Representations

Another usage of IP involves representation of the multiplication of zero-one variables. A term involving the product of two zero-one variables would equal one when both integer variables equal one and zero otherwise. Suppose  $Z$  equals the product of two zero-one variables  $X_1$  and  $X_2$ ,

$$Z = X_1 X_2.$$

We may replace this term by introducing  $Z$  as a zero-one variable as follows:

$$\begin{aligned} Z &\geq X_1 + X_2 - 1 \\ 2Z &\leq X_1 + X_2 \\ Z, X_1, X_2 &\in \{0, 1\} \end{aligned}$$

The first constraint requires that  $Z + 1$  be greater than or equal to  $X_1 + X_2$ . Thus,  $Z$  is forced to equal 1 if both  $X_1$  and  $X_2$  equal one. The second constraint requires  $2Z$  to be less than or equal to  $X_1 + X_2$ . This permits  $Z$  to be nonzero only when both  $X_1$  and  $X_2$  equal one. Thus,  $Z$  will equal zero if either of the variables equal zero and will equal one when both  $X_1$  and  $X_2$  are one. One may not need both constraints, for example, when  $Z$  appears with positive returns in a profit maximizing objective function the first constraint could be dropped, although as discussed later it can be important to keep many constraints when doing applied IP.

### 15.1.6 Approximation of Nonlinear Functions

IP is useful for approximating nonlinear functions, which cannot be approximated with linear programming i.e., functions with increasing marginal revenue or decreasing marginal cost. (LP step approximations cannot adequately approximate this; the resultant objective function is not concave.) One can formulate an IP to require the approximating points to be adjacent making the formulation work appropriately. If one has four step variables, an adjacency restriction can be imposed as follows:



A graph of this situation is given by Figure 15.1. The diamonds in the graph represent the integer points, which are the potential integer solutions. Obviously the feasible integer solution points fall below or on the constraints while simultaneously being above or on the X and Y axes. For this example the optimal solution is probably not on the constraint boundaries (i.e.  $X=Y$  may be optimal), much less at the constraint intersections. This introduces the principal difficulty in solving IP problems. There is no particular location for the potential solutions. Thus, while the equivalent LP problem would have four possible solutions (each feasible extreme point and the origin), the IP problem has an unknown number of possible solutions. No general statement can be made about the location of the solutions.

A second difficulty is that, given any two feasible solutions, all the points in between are not feasible (i.e., given [3 3] and [2 4], all points in between are non-integer). Thus, one cannot move freely within the IP area maintaining problem feasibility, rather one must discover the IP points and move totally between them.

Thirdly, it is difficult to move between feasible points. This is best illustrated by a slightly different example. Suppose we have the constraints

$$\begin{aligned} X_1 & \leq 10 \\ X_2 & \leq 54 \\ X_1 & \leq 10X_2 \end{aligned}$$

where  $X_1$  and  $X_2$  are nonnegative integer variables. A graph of the solution space appears in Figure 15.2. Note here the interrelationship of the feasible solutions do not exhibit any set patterns. In the first graph one could move between the extreme feasible solutions by moving over one and down one. In Figure 15.2, different patterns are involved. A situation which greatly hampers IP algorithms is that it is difficult to maintain feasibility while searching for optimality. Further, in Figure 15.2, rounding the continuous solution at say (4.6, 8.3) leads to an infeasible integer solution (at 5, 8).

Another cause of solution difficulties is the discontinuous feasible region. Optimization theory traditionally has been developed using calculus concepts. This is illustrated by the LP reduced cost ( $Z_j - C_j$ ) criteria and by the Kuhn-Tucker theory for nonlinear programming. However, in an IP setting, the discontinuous feasible region does not allow the use of calculus. There is no neighborhood surrounding a feasible point that one can use in developing first derivatives. Marginal revenue-marginal cost concepts are not readily usable in an IP setting. There is no decision rule that allows movement to higher valued points. Nor can one develop a set of conditions (i.e., Kuhn-Tucker conditions) which characterize optimality.

In summary, IP feasible regions contain a finite number of solution alternatives, however, there is no rule for either the number of feasible solution alternatives or where they are located. Solution points may be on the boundary of the constraints at the extreme points or interior to the feasible region. Further, one cannot easily move between feasible points. One cannot derive marginal revenue or marginal cost

information to help guide the solution search process and to more rapidly enumerate solutions. This makes IP's more difficult to solve. There are a vast number of solutions, the number of which to be explored is unknown. Most IP algorithms enumerate (either implicitly or explicitly) all possible integer solutions requiring substantial search effort. The binding constraints are not binding in the linear programming sense. Interior solutions may occur with the constraint restricting the level of the decision variables.

### 15.2.1 Extension to Mixed Integer Feasible Regions

The above comments reference pure IP. Many of them, however, are also relevant to mixed IP. Consider a graph (Figure 15.3) of the feasible region to the constraints

$$\begin{array}{rcl}
 2X_1 & \% & 3X_2 \# & 16 \\
 3X_1 & \% & 2X_2 \# & 16 \\
 X_1 & & \$ & 0 \text{ integer} \\
 & & X_2 \$ & 0
 \end{array}$$

The feasible region is a set of horizontal lines for  $X_2$  at each feasible integer value of  $X_1$ . This yields a discontinuous area in the  $X_1$  direction but a continuous area in the  $X_2$  direction. Thus, mixed integer problems retain many of the complicating features of pure integer problems along with some of the niceties of LP problem feasible regions.

### 15.3 Sensitivity Analysis and Integer Programming

The reader may wonder, given the concentration of this book on problem formulation and solution interpretation, why so little was said above about integer programming duality and associated valuation information. There are several reasons for this lack of treatment. Duality is not a well-defined subject in the context of IP. Most LP and NLP duality relationships and interpretations are derived from the calculus constructs underlying Kuhn-Tucker theory. However, calculus cannot be applied to the discontinuous integer programming feasible solution region. In general, dual variables are not defined for IP problems, although the topic has been investigated (Gomory and Baumol; Williams, 1980). All one can generally state is that dual information is not well defined in the general IP problem. However, there are two aspects to such a statement that need to be discussed.

First, most commonly used algorithms printout dual information. But the dual information is often influenced by constraints which are added during the solution process. Most solution approaches involve the addition of constraints to redefine the feasible region so that the integer solutions occur at extreme points (see the discussions of algorithms below). Thus, many of the shadow prices reported by IP codes are not relevant to the original problem, but rather are relevant to a transformed problem. The principal difficulty with these dual prices is that the set of transformations is not unique, thus the new information is not unique or complete (see the discussion arising in the various duality papers such as that by Gomory and

Baumol or those referenced in von Randow). Thus, in many cases, the IP shadow price information that appears in the output should be ignored.

Second, there does seem to be a major missing discussion in the literature. This involves the reliability of dual variables when dealing with mixed IP problems. It would appear to follow directly from LP that mixed IP shadow prices would be as reliable as LP shadow prices if the constraints right hand sides are changed in a range that does not imply a change in the solution value of an integer variable. The dual variables from the constraints which involve only continuous variables would appear to be most accurate. Dual variables on constraints involving linkages between continuous and integer variable solution levels would be less accurate and constraints which only involve integer variables would exhibit inaccurate dual variables. This would be an interesting topic for research as we have not discovered it in the IP literature.

The third dual variable comment regards "binding" constraints. Consider Figure 15.1. Suppose that the optimal solution occurs at  $X=3$  and  $Y=3$ . Note that this point is strictly interior to the feasible region. Consequently, according to the complementary slackness conditions of LP, the constraints would have zero dual variables. On the other hand, if the first constraint was modified so that its right hand side was more than 17, the solution value could move to  $X=4$  and  $Y=3$ . Consequently, the first constraint is not strictly binding but a relaxation of its right hand side can yield to an objective function increase. Therefore, conceptually, it has a dual variable. Thus, the difficulty with dual variables in IP is that they may be nonzero for nonbinding constraints.

#### **15.4 Solution Approaches to Integer Programming Problems**

IP problems are notoriously difficult to solve. They can be solved by several very different algorithms. Today, algorithm selection is an art as some algorithms work better on some problems. We will briefly discuss algorithms, attempting to expose readers to their characteristics. Those who wish to gain a deep understanding of IP algorithms should supplement this chapter with material from the literature (e.g., see Balinski or Bazaraa and Jarvis; Beale (1965,1977); Garfinkel and Nemhauser; Geoffrion and Marsten; Hammer et al.; Hu; Plane and McMillan; Salkin (1975b); Taha (1975); von Randow; Zions; Nemhauser; and Woolsey). Consultation with experts, solution experimentation and a review of the literature on solution codes may also be necessary when one wishes to solve an IP problem.

Let us develop a brief history of IP solution approaches. LP was invented in the late 1940's. Those examining LP relatively quickly came to the realization that it would be desirable to solve problems which had some integer variables (Dantzig, 1960). This led to algorithms for the solution of pure IP problems. The first algorithms were cutting plane algorithms as developed by Dantzig, Fulkerson and Johnson (1954) and Gomory (1958, 1960, 1963). Land and Doig subsequently introduced the branch and bound algorithm. More recently, implicit enumeration (Balas), decomposition (Benders), lagrangian relaxation (Geoffrion, 1974) and heuristic (Zanakis and Evans) approaches have been used. Unfortunately,

after 20 years of experience involving literally thousands of studies (see Von Randow) none of the available algorithms have been shown to perform satisfactorily for all IP problems. However, certain types of algorithms are good at solving certain types of problems. Thus, a number of efforts have concentrated on algorithmic development for specially structured IP problems. The most impressive recent developments involve exploitation of problem structure. The section below briefly reviews historic approaches as well as the techniques and successes of structural exploitation. Unfortunately, complete coverage of these topics is far beyond the scope of this text. In fact, a single, comprehensive treatment also appears to be missing from the general IP literature, so references will be made to treatments of each topic.

There have been a wide variety of approaches to IP problems. The ones that we will cover below include Rounding, Branch and Bound, Cutting Planes, Lagrangian Relaxation, Benders Decomposition, and Heuristics. In addition we will explicitly deal with Structural Exploitation and a catchall other category.

### 15.4.1 Rounding

Rounding is the most naive approach to IP problem solution. The rounding approach involves the solution of the problem as a LP problem followed by an attempt to round the solution to an integer one by: a) dropping all the fractional parts; or b) searching out satisfactory solutions wherein the variable values are adjusted to nearby larger or smaller integer values. Rounding is probably the most common approach to solving IP problems. Most LP problems involve variables with fractional solution values which in reality are integer (i.e., chairs produced, chickens cut up). Fractional terms in solutions do not make strict sense, but are sometimes acceptable if rounding introduces a very small change in the value of the variable (i.e. rounding 1003421.1 to 1003421 or even 1003420 is probably acceptable).

There is, however, a major difficulty with rounding. Consider the example

$$\begin{array}{rcl} X_1 & + & 7X_2 \leq 22.5 \\ X_1 & + & 10X_2 \leq 54 \\ X_1, X_2 & & \geq 0 \text{ and integer} \end{array}$$

as graphed in Figure 15.2. In this problem rounding would produce a solution outside the feasible region.

In general, rounding is often practical, but it should be used with care. One should compare the rounded and unrounded solutions to see whether after rounding: a) the constraints are adequately satisfied; and b) whether the difference between the optimal LP and the post rounding objective function value is reasonably small. If so IP is usually not cost effective and the rounded solution can be used. On the other hand, if one finds the rounded objective function to be significantly altered or the constraints violated from a pragmatic viewpoint, then a formal IP exercise needs to be undertaken.

### 15.4.2 Cutting Planes

The first formal IP algorithms involved the concept of cutting planes. Cutting planes remove part of the feasible region without removing integer solution points. The basic idea behind a cutting plane is that the optimal integer point is close to the optimal LP solution, but does not fall at the constraint intersection so additional constraints need to be imposed. Consequently, constraints are added to force the noninteger LP solution to be infeasible without eliminating any integer solutions. This is done by adding a constraint forcing the nonbasic variables to be greater than a small nonzero value. Consider the following integer program:

$$\begin{aligned}
 &\text{Maximize} && X_1 & \% & X_2 \\
 &&& 2X_1 & \% & 3X_2 & \# & 16 \\
 &&& 3X_1 & \% & 2X_2 & \# & 16 \\
 &&& X_1, & & X_2 & \$ & 0 \text{ and integer}
 \end{aligned}$$

The optimal LP solution tableau is

	$X_1$	$X_2$	$S_1$	$S_2$	b
obj	1.4	1	0	0	
$X_1$	1	0	.6	.4	3.2
$X_2$	0	1	.4	.6	3.2
$Z_j \& C_j$	0	0	.2	.2	6.4

which has  $X_1=X_2=3.2$  which is noninteger. The simplest form of a cutting plane would be to require the sum of the nonbasic variables to be greater than or equal to the fractional part of one of the variables. In particular, generating a cut from the row where  $X_1$  is basic allows a constraint to be added which required that  $0.6 S_1 - .4 S_2 \geq 0.2$ . The cutting plane algorithm continually adds such constraints until an integer solution is obtained.

Much more refined cuts have been developed. The issue is how should the cut constraint be formed. Methods for developing cuts appear in Gomory (1958, 1960, 1963).

Several points need to be made about cutting plane approaches. First, many cuts may be required to obtain an integer solution. For example, Beale (1977) reports that a large number of cuts is often required (in fact often more are required than can be afforded). Second, the first integer solution found is the optimal solution. This solution is discovered after only enough cuts have been added to yield an integer solution. Consequently, if the solution algorithm runs out of time or space the modeler is left without an acceptable solution (this is often the case). Third, given comparative performance vis-a-vis other algorithms, cutting plane approaches have faded in popularity (Beale,1977).

### 15.4.3 Branch and Bound

The second solution approach developed was the branch and bound algorithm. Branch and bound, originally introduced by Land and Doig, pursues a divide-and-conquer strategy. The algorithm starts with a



LP solution and also imposes constraints to force the LP solution to become an integer solution much as do cutting planes. However, branch and bound constraints are upper and lower bounds on variables. Given the noninteger optimal solution for the example above (i.e.,  $X_1 = 3.2$ ), the branch and bound algorithm would impose constraints requiring  $X_1$  to be at or below the adjacent integer values around 3.2; i.e.,  $X_1 \leq 3$  and  $X_1 \geq 4$ . This leads to two disjoint problems, i.e.,

$$\left[ \begin{array}{rcl} \text{Maximize} & 1.4X_1 & + X_2 \\ & 2X_1 & + 3X_2 \leq 16 \\ & 3X_1 & + 2X_2 \leq 16 \\ & X_1 & \leq 3 \\ & X_1, & X_2 \geq 0 \end{array} \right] \text{ and } \left[ \begin{array}{rcl} \text{Maximize} & 1.4X_1 & + X_2 \\ & 2X_1 & + 3X_2 \leq 16 \\ & 3X_1 & + 2X_2 \leq 16 \\ & X_1 & \geq 4 \\ & X_1, & X_2 \geq 0 \end{array} \right]$$

The branch and bound solution procedure generates two problems (branches) after each LP solution. Each problem excludes the unwanted noninteger solution, forming an increasingly more tightly constrained LP problem. There are several decisions required. One must both decide which variable to branch upon and which problem to solve (branch to follow). When one solves a particular problem, one may find an integer solution. However, one cannot be sure it is optimal until all problems have been examined. Problems can be examined implicitly or explicitly. Maximization problems will exhibit declining objective function values whenever additional constraints are added. Consequently, given a feasible integer solution has been found, then any solution, integer or not, with a smaller objective function value cannot be optimal, nor can further branching on any problem below it yield a better solution than the incumbent (since the objective function will only decline). Thus, the best integer solution found at any stage of the algorithm provides a bound limiting the problems (branches) to be searched. The bound is continually updated as better integer solutions are found.

The problems generated at each stage differ from their parent problem only by the bounds on the integer variables. Thus, a LP algorithm which can handle bound changes can easily carry out the branch and bound calculations.

The branch and bound approach is the most commonly used general purpose IP solution algorithm (Beale, 1977; Lawler and Wood). It is implemented in many codes (e.g., OSL, LAMPS, and LINDO) including all of those interfaced with GAMS. However, its use can be expensive. The algorithm does yield intermediate solutions which are usable although not optimal. Often the branch and bound algorithm will come up with near optimal solutions quickly but will then spend a lot of time verifying optimality. Shadow prices from the algorithm can be misleading since they include shadow prices for the bounding constraints.

A specialized form of the branch and bound algorithm for zero-one programming was developed by Balas. This algorithm is called implicit enumeration. This method has also been extended to the mixed integer case as implemented in LINDO (Schrage, 1981b).

### 15.4.5 Lagrangian Relaxation

Lagrangian relaxation (Geoffrion (1974), Fisher (1981, 1985)) is another area of IP algorithmic development. Lagrangian relaxation refers to a procedure in which some of the constraints are relaxed into the objective function using an approach motivated by Lagrangian multipliers. The basic Lagrangian Relaxation problem for the mixed integer program:

$$\begin{aligned} & \text{Maximize } CX \% FY \\ & \text{s.t. } AX \% GY \# b \\ & \quad DX \% HY \# e \\ & \quad X \$ 0, Y \$ 0 \text{ and integer,} \end{aligned}$$

involves discovering a set of Lagrange multipliers for some constraints and relaxing that set of constraints into the objective function. Given that we choose to relax the second set of constraints using lagrange multipliers (8) the problem becomes

$$\begin{aligned} & \text{Maximize } CX \% FY \& \delta(DH \% HY \& e) \\ & \text{s.t. } AX \% GY \# b \\ & \quad X \$ 0, Y \$ 0 \text{ and integer,} \end{aligned}$$

The main idea is to remove difficult constraints from the problem so the integer programs are much easier to solve. IP problems with structures like that of the transportation problem can be directly solved with LP. The trick then is to choose the right constraints to relax and to develop values for the lagrange multipliers (8<sub>k</sub>) leading to the appropriate solution.

Lagrangian Relaxation has been used in two settings: 1) to improve the performance of bounds on solutions; and 2) to develop solutions which can be adjusted directly or through heuristics so they are feasible in the overall problem (Fisher (1981, 1985)). An important Lagrangian Relaxation result is that the relaxed problem provides an upper bound on the solution to the unrelaxed problem at any stage. Lagrangian Relaxation has been heavily used in branch and bound algorithms to derive upper bounds for a problem to see whether further traversal down that branch is worthwhile.

Lagrangian Relaxation has been applied extensively. There have been studies of the travelling salesman problem (Bazaraa and Goode), power generation systems (Muckstadt and Koenig); capacitated location problem (Cornuejols, et al.); capacitated facility location problem (Geoffrion and McBride); and generalized assignment problem (Ross and Soland). Fisher (1981,1985) and Shapiro (1979a) present

survey articles.

### 15.4.6 Benders Decomposition

Another algorithm for IP is called Benders Decomposition. This algorithm solves mixed integer programs via structural exploitation. Benders developed the procedure, thereafter called Benders Decomposition, which decomposes a mixed integer problem into two problems which are solved iteratively - an integer master problem and a linear subproblem.

The success of the procedure involves the structure of the subproblem and the choice of the subproblem. The procedure can work very poorly for certain structures. (e.g. see McCarl, 1982a or Bazarra, Jarvis and Sherali.)

A decomposable mixed IP problem is:

$$\begin{array}{llll} \text{Maximize} & FX & \% & CZ \\ \text{s.t.} & GX & & \# b_1 \\ & HX & \% & AZ \# b_2 \\ & & & DZ \# b_3 \end{array}$$

Development of the decomposition of this problem proceeds by iteratively developing feasible points  $X^*$  and solving the subproblem:

$$\begin{array}{ll} \text{Maximize} & CZ \\ \text{s.t.} & AZ \# b_2 \ \& \ HX( \quad (") \\ & DZ \# b_3 \quad (()) \\ & Z \ \$ \ 0 \end{array}$$

Solution to this subproblem yields the dual variables in parentheses. In turn a "master" problem is formed as follows

$$\begin{array}{ll} \text{Maximize} & FX \% Q \\ X, \ " , \ ( , \ Q & \\ & Q \# \ " (b_2 \ \& \ HX) \% \ ( b_3 \ i \ ' \ 1, 2, \dots, p \\ & GX \# b_1 \\ & X \text{ integer} \\ & Q \begin{array}{l} \geq 0 \\ < \end{array} \end{array}$$

This problem contains the dual information from above and generates a new  $X$  value. The constraint involving  $Q$  gives a prediction of the subproblem objective function arising from the dual variables from the  $i^{\text{th}}$  previous guess at  $X$ . In turn, this problem produces a new and better guess at  $X$ . Each iteration adds a

constraint to the master problem. The objective function consists of  $FX + Q$ , where  $Q$  is an approximation of  $CZ$ . The master problem objective function therefore constitutes a monotonically nonincreasing upper bound as the iterations proceed. The subproblem objective function ( $CZ$ ) at any iteration plus  $FX$  can be regarded as a lower bound. The lower bound does not increase monotonically. However, by choosing the larger of the current candidate lower bound and the incumbent lower bound, a monotonic nondecreasing sequence of bounds is formed. The upper and lower bounds then give a monotonically decreasing spread between the bounds. The algorithm user may stop the solution process at an acceptably small bound spread. The last solution which generated a lower bound is the solution which is within the bound spread of the optimal solution. The form of the overall problem guarantees global optimality in most practical cases. Global optimality will occur when all possible  $X$ 's have been enumerated (either implicitly or explicitly). Thus, Benders decomposition convergence occurs when the difference between the bounds is driven to zero. When the problem is stopped with a tolerance, the objective function will be within the tolerance, but there is no relationship giving distance between the variable solutions found and the true optimal solutions for the variables. (i.e., the distance of  $Z^*$  and  $X^*$  from the true optimal  $Z$ 's and  $X$ 's). Convergence will occur in a practical setting only if for every  $X$  a relevant set of dual variables is returned. This will only be the case if the subproblem is bounded and has a feasible solution for each  $X$  that the master problem yields. This may not be generally true; artificial variables may be needed.

However, the boundedness and feasibility of the subproblem says nothing about the rate of convergence. A modest sized linear program will have many possible (thousands, millions) extreme point solutions. The real art of utilizing Benders decomposition involves the recognition of appropriate problems and/or problem structures which will converge rapidly. The general statements that can be made are:

1. The decomposition method does not work well when the  $X$  variables chosen by the master problem do not yield a feasible subproblem. Thus, the more accurately the constraints in the master problem portray the conditions of the subproblem, the faster will be convergence. (See Geoffrion and Graves; Danok, McCarl and White (1978); Polito; Magnanti and Wong; and Sherali for discussion.)
2. The tighter (more constrained) the feasible region of the master problem the better. (See Magnanti and Wong; and Sherali.)
3. When possible, constraints should be entered in the master problem precluding feasible yet unrealistic (suboptimal) solutions to the overall problem. (See the minimum machinery constraints in Danok, McCarl and White, 1978.)

The most common reason to use Benders is to decompose large mixed integer problem into a small, difficult master problem and a larger simple linear program. This allows the solution of the problem by two pieces of software which individually would not be adequate for the overall problem but collectively

are sufficient for the resultant pieces. In addition, the decomposition may be used to isolate particular easy-to-solve subproblem structures (see the isolation of transportation problems as in Geoffrion and Graves or Hilger et al.). Finally, multiple levels of decomposition may be done in exploiting structure (see Polito).

#### **15.4.7 Heuristics**

Many IP problems are combinatorial and difficult to solve by nature. In fact, the study of NP complete problems (Papadimitrou and Steiglitz) has shown extreme computational complexity for problems such as the traveling salesman problem. Such computational difficulties have led to a large number of heuristics. These heuristics (following Zanakis and Evans) are used when: a) the quality of the data does not merit the generation of exact optimal solutions; b) a simplified model has been used, and/or c) when a reliable exact method is not available, computationally attractive, and/or affordable. Arguments for heuristics are also presented regarding improving the performance of an optimizer where a heuristic may be used to save time in a branch and bound code, or if the problem is repeatedly solved. Many IP heuristics have been developed, some of which are specific to particular types of problems. For example, there have been a number of traveling salesman problem heuristics as reviewed in Golden et al. Heuristics have been developed for general 0-1 programming (Senju and Toyoda; Toyoda) and general IP (Glover; Kochenberger, McCarl, and Wyman), as well as 0-1 polynomial problems (Granot). Zanakis and Evans review several heuristics, while Wyman presents computational evidence on their performance. Generally, heuristics perform well on special types of problems, quite often coming up with errors of smaller than two percent. Zanakis and Evans; and Wyman both provide discussions of selections of heuristics vis-a-vis one another and optimizing methods. Heuristics also do not necessarily reveal the true optimal solution, and in any problem, one is uncertain as to how far one is from the optimal solution although the Lagrangian Relaxation technique can make bounding statements.

#### **15.4.8 Structural Exploitation**

Years of experience and thousands of papers on IP have indicated that general-purpose IP algorithms do not work satisfactorily for all IP problems. The most promising developments in the last several years have involved structural exploitation, where the particular structure of a problem has been used in the development of the solution algorithm. Such approaches have been the crux of the development of a number of heuristics, the Benders Decomposition approaches, Lagrangian Relaxation and a number of problem reformulation approaches. Specialized branch and bound algorithms adapted to particular problems have also been developed (Fuller, Randolph and Klingman; Glover et al. ,1978). The application of such algorithms has often led to spectacular results, with problems with thousands of variables being solved in seconds of computer time (e.g., see the computational reports in Geoffrion and Graves; Zanakis; and the references in Fisher, 1985). The main mechanisms for structural exploitation are to develop an

algorithm especially tuned to a particular problem or, more generally, to transform a problem into a simpler problem to solve.

#### **15.4.9 Other Solution Algorithms and Computer Algorithms**

The above characterization of solution algorithms is not exhaustive. A field as vast as IP has spawned many other types of algorithms and algorithmic approaches. The interested reader should consult the literature reviews in von Randow; Geoffrion (1976); Balinski; Garfinkel and Nemhauser; Greenberg (1971); Woolsey; Shapiro (1979a, 1979b); and Cooper as well as those in textbooks.

#### **15.5 The Quest for Global Optimality: Non-Convexity**

Most of the IP solution material, as presented above, showed the IP algorithms as involving some sort of an iterative search over the feasible solution region. All possible solutions had to be either explicitly or implicitly enumerated. The basic idea behind most IP algorithms is to search out the solutions. The search process involves implicit or explicit enumeration of every possible solution. The implicit enumeration is done by limiting the search based on optimality criterion (i.e., that solutions will not be examined with worse objective functions than those which have been found). The enumeration concept arises because of the nonconvex nature of the constraint set; in fact, in IP it is possible to have a disjoint constraint set. For example, one could implement an IP problem with a feasible region requiring  $X$  to be either greater than 4 or less than 5. Thus, it is important to note that IP algorithms can guarantee global optimality only through an enumerative search. Many of the algorithms also have provisions where they stop depending on tolerances. These particular algorithms will only be accurate within the tolerance factor specified and may not reveal the true optimal solution.

#### **15.6 Formulation Tricks for Integer Programming - Add More Constraints**

IP problems, as alluded to above, involve enumerative searches of the feasible region in an effort to find the optimal IP solutions. Termination of a direction of search occurs for one of three reasons: 1) a solution is found; 2) the objective function is found to go below some certain value, or 3) the direction is found to possess no feasible integer solutions. This section argues that this process is speeded up when the modeler imposes as many reasonable constraints as possible for defining the feasible and optimal region. Reasonable means that these constraints are not redundant, each uniquely helping define and reduce the size of the feasible solution space.

LP algorithms are sensitive to the number of constraints. Modelers often omit or eliminate constraints when it appears the economic actions within the model will make these constraints unnecessary. However, in IP, it is often desirable to introduce constraints which, while appearing unnecessary, can greatly decrease solution time. In order to clarify this argument, three cases are cited from our experiences with the solution of IP models.

In the first example, drawn from Danok's masters thesis (1976), Danok was solving a mixed IP problem of machinery selection. The problem was solved using Benders decomposition, in which the integer program for machinery selection was solved iteratively in association with a LP problem for machinery use. Danok solved two versions. In the first, the machinery items were largely unconstrained. In the second, Danok utilized the amount of machinery bought in the LP solution as a guide in imposing constraints on the maximum and minimum amount of types of machinery. Danok constrained the solution so that no more than 50 percent more machinery could be purchased than that utilized in the optimal LP solution (i.e., ignoring the integer restrictions). The solution time reduction between the formulations were dramatic. The model with the extra constraints solved in less than 10 percent of the computer time. However, the solutions were identical and far away from the LP derived constraints. Thus, these constraints greatly reduced the number of solutions which needed to be searched through, permitting great efficiencies in the solution process. In fact, on the larger Danok problem, the amount of computer time involved was considerable (over 1,000 seconds per run) and these constraints allowed completion of the research project.

The second example arose in Polito's Ph.D. thesis. Polito was solving a warehouse location type problem and solved two versions of the problem (again with Benders decomposition). In the first version, constraints were not imposed between the total capacity of the plants constructed and the demand. In the second problem, the capacity of the plants located were required to be greater than or equal to the existing demand. In the first problem, the algorithm solved in more than 350 iterations; in the second problem only eight iterations were required.

The third example arises in Williams (1978a or 1978b) wherein constraints like

$$Y_1 \% Y_2 \& Md \# 0$$

including the indicator variable d, are replaced with

$$Y_1 \& Md \# 0$$

$$Y_2 \& Md \# 0$$

which has more constraints. The resultant solution took only 10 percent of the solution time.

In all cases the imposition of seemingly obvious constraints, led to great efficiencies in solution time. Thus, the integer programmer should use constraints to tightly define the feasible region. This eliminates possible solutions from the enumeration process.

### 15.7 IP Solutions and GAMS

The solution of integer programs with GAMS is achieved basically by introducing a new class of variable declaration statements and by invoking an IP solver. The declaration statement identifies selected variables to either be `BINARY` (zero one) or `INTEGER`. In turn, the model is solved by utilizing a solved statement which says "`USING MIP`". Table 1 shows an example formulation and Table 2 the GAMS input

string. This will cause GAMS to use the available integer solvers. Currently the code ZOOM is distributed with the student version, but we do not recommend ZOOM for practical integer programming problems. Those wishing to solve meaningful problems should use OSL, LAMPS, XA, CPLEX or one of the other integer solvers.



## References

- Balas, E. "An Additive Algorithm for Solving Linear Programs with Zero-One Variables." Operations Research 13(1965):517-546.
- Balinski, M.L. "Integer Programming Methods, Uses, Computation." Management Science. 12:253-313.
- Bazaraa, M.S. and J.J. Goode. "The Traveling Salesman Problem: A Duality Approach". Mathematical Programming. 13(1977):221-237.
- Bazaraa, M.S. and J. Jarvis. Linear Programming and Network Flows. John Wiley & Sons, 1977.
- Beale, E.M.L. "Survey of Integer Programming." Operation Research Quarterly 16:2(1965):219-228.
- \_\_\_\_\_. "Integer Programming," in D. Jacobs (ed.) The State of the Art in Numerical Analysis. Academic Press, New York, NY, 1977.
- Benders, J.F. "Partitioning Procedures for Solving Mixed-Variables Programming Problems." Numerical Methods. 4(1962):239-252.
- Cooper, M.W. "A Survey of Methods for Pure Nonlinear Integer Programming." Management Science. 27(1981):353-361.
- Cornuejols, G., M.L. Fisher, and G.L. Nemhauser. "Location of Bank Accounts to Optimize Float: Analytic Study of Exact and Approximate Algorithms." Management Science. 23(1977):789-810.
- Danok, A.B. "Machinery Selection and Resource Allocation on a State Farm in Iraq." M.S. thesis, Dept. of Agricultural Economics, Purdue University, 1976.
- Danok, A.B., B.A. McCarl, and T.K. White. "Machinery Selection and Crop Planning on a State Farm in Iraq." American Journal of Agricultural Economics. 60(1978):544-549.
- \_\_\_\_\_. "Machinery Selection Modeling: Incorporation of Weather Variability." American Journal of Agricultural Economics. 62(1980):700-708.
- Dantzig, G.B. "Discrete Variable Extremum Problems." Operations Research. 5(1957):266-277.
- \_\_\_\_\_. "Notes on Solving Linear Programs in Integers." Naval Research Logistics Quarterly. 6(1959):75-76.
- \_\_\_\_\_. "On the Significance of Solving Linear Programming Problems with Some Integer Variables." Econometrica 28(1960):30-44.
- \_\_\_\_\_. Linear Programming and Extensions. Princeton University Press. Princeton, New Jersey, 1963.
- Dantzig, G.B., D.R. Fulkerson, and S.M. Johnson. "Solution of a Large Scale Traveling Salesman Problem". Operations Research. 2(1954):393-410.
- Driebeck, N.J. "An Algorithm for the Solution of Mixed Integer Programming Problems." Management Science. 12(1966):576-587.
- Fisher, M.L. "Worst Case Analysis of Heuristic Algorithms." Management Science. 26(1980):1-17.
- \_\_\_\_\_. "The Lagrangian Relaxation Method for Solving Integer Programming Problems." Management Science 27(1981):1-18.
- \_\_\_\_\_. "An Applications Oriented Guide to Lagrangian Relaxation." Interfaces (forthcoming), 1985.
- Fisher, M.L., A.J. Greenfield, R. Jaikumar, and J.T. Lester III. "A Computerized Vehicle Routing Application." Interfaces 12(1982):42-52.
- Fuller, W.W., P. Randolph, and D. Klingman. "Optimizing Subindustry Marketing Organizations: A Network Analysis Approach." American Journal of Agricultural Economics. 58(1976):425-436.
- Garfinkel, R.S. and G.L. Nemhauser. Integer Programming. New York: John Wiley and Sons, 1972.

- Geoffrion, A.M. "Integer Programming by Implicit Enumeration and Balas' Method." SIAM Review of Applied Mathematics. 9(1969):178-190.
- \_\_\_\_\_. "Generalized Benders Decomposition." Journal of Optimization Theory and Application, 10(1972):237-260.
- \_\_\_\_\_. "Lagrangian Relaxation and its Uses in Integer Programming." Mathematical Programming Study, 2(1974):82-114.
- \_\_\_\_\_. "A Guided Tour of Recent Practical Advances in Integer Linear Programming." Omega 4(1976):49-57.
- Geoffrion, A.M. and G.W. Graves. "Multicommodity Distribution System Design by Bender's Decomposition." Management Science. 23(1977):453-466.
- Geoffrion, A.M. and R.E. Marsten. "Integer Programming: A Framework and State-of-the-Art Survey." Management Science. 18(1972):465-491.
- Geoffrion, A.M. and R. McBride. "Lagrangian Relaxation Applied to Capacitated Facility Location Problems." American Institute of Industrial Engineers Transactions. 10(1978):40-47.
- Glover, F. Heuristics for Integer Programming Using Surrogate Constraints." Decision Sciences 8(1977):156-166.
- Glover, F., J. Hultz, D. Klingman and J. Stutz. "Generalized Networks: A Fundamental Computer-Based Planning Tool." Management Science. 24(1978):1209-20.
- Golden, B., L. Bodin, T. Doyle and W. Stewart. "Approximate Traveling Salesman Algorithms." Operations Research. 28(1980):694-711.
- Gomory, R.E. "Outline of an Algorithm for Integer Solutions to Linear Programs." Bulletin of the American Mathematics Society. 64(1958):275-278.
- \_\_\_\_\_. "Solving Linear Programming Problems in Integers." 10th Proceedings, Symposium on Applied Mathematics sponsored by the American Mathematics Society. (R.B. Bellman and M. Hall, Jr., eds.), 1960:211-216.
- \_\_\_\_\_. 1963. "An Algorithm for Integer Solutions to Linear Programs." In Recent Advances in Mathematical Programming. (R.L. Graves and P. Wolfe, eds.). McGraw-Hill, New York, 1963:269,302.
- Gomory, R.E. and W.J. Baumol. "Integer Programming and Pricing." Econometrica 28(1960):521-550.
- Granot, F. "Efficient Heuristics Algorithms for Postive 0-1 Polynomial Programming Problems." Management Science. 28(1982):829-836.
- Graves, S.C. "Using Lagrangian Techniques to Solve Hierarchical Production Planning Problems." Management Science 28(1982):260-275.
- Greenberg,H. Integer Programming.Academic Press, Inc., New York, 1971.
- Hammer, P.L., et al. 1977. Studies in Integer Programming. North Holland, Inc., New York, 1977.
- Hilger, D.A., B.A. McCarl, and J.W. Uhrig. "Facilities Location: The Case of Grain Subterminals." American Journal of Agricultural Economics 59(1977):674-682.
- Hu, T.C. Integer Programming and Network Flows. Addison-Wesley Publishing Company, Reading, MA. 1969.
- Kochenberger, G.A., B.A. McCarl, and F.P. Wyman. "A Heuristic for General Integer Programming." Decision Sciences. 5(1974):36-44.
- Land, A.H. and A.G. Doig. "An Automatic Method for Solving Discrete Programming Problems." Econometrica 28(1960):497-520.
- Lawler, E.L. and D.E. Wood. "Branch and Bound Methods: A Survey." Operations Research 1966:669-719.

- Magnanti, T.L. and R.T. Wong. "Accelerating Benders Decomposition: Algorithmic Enhancements and Model Selection Criteria." Operations Research, 29(1981):464-484.
- Markowitz, H.M. and A. Manne. "On the Solution of Discrete Programming Problems." Econometrica. 25(1975):84-110.
- McCarl, B.A. 1982. Benders Decomposition. Purdue University Agricultural Experiment Station Bulletin 361, 1982.
- Papadimitrou, C.H. and K. Steiglitz. Combinatorial Optimization: Algorithms and Complexity. Prentice-Hall, Englewood Cliffs, NJ, 1982.
- Plane, D.R. and C. McMillan, Jr. Discrete Optimization--Integer Programming and Network Analysis for Management Decisions. Prentice-Hall, Inc., Englewood Cliffs, NJ:1971.
- Polito, J. "Distribution Systems Planning in a Price Responsive Environment." Unpublished Ph.D. Dissertation, Purdue Univ. West Lafayette, Ind. 1977.
- Ross, G.T. and R.M. Soland. "A Branch and Bound Algorithm for the Generalized Assignment Problem." Mathematical Programming. 8(1975):91-103.
- Salkin, H. "The Knapsack Problem." Naval Research Logistics Quarterly. 22(1975):127-155
- \_\_\_\_\_. Integer Programming. Addison-Wesley, Reading, Mass, 1975.
- Schrage, L.E. Linear Programming Models with Lindo. The Scientific Press, Palo Alto, Ca, 1981.
- Senju, S. and Y. Toyoda. "An Approach to Linear Programming with 0-1 Variables." Management Science. 15(1968):B196-B207.
- Shapiro, J.F. "A Survey of Lagrangian Techniques for Discrete Optimization." Annals of Discrete Mathematics. 5(1979a):113-38.
- \_\_\_\_\_. Mathematical Programming: Structure and Algorithms. John Wiley & Sons. New York, 1979b.
- Sherali, H. "Expedients for Solving Some Specially Structured Mixed- Integer Programs." Naval Research Logistics Quarterly. 28(1981):447-62.
- Taha, H.A. Integer Programming - Theory, Applications, and Computations Academic Press, New York, 1975.
- Tonge, F.M. "The Use of Heuristic Programming In Management Science." Management Science. 7(1961):231-37.
- Toyoda, S. "A Simplified Algorithm for Obtaining Approximate Solutions to 0-1 Programming Problems." Management Science. 21(1975):1417-27.
- von Randow, R. Integer Programming and Related Areas - A Classified Bibliography. 1978-1981. Springer-Verlag, New York, 1982.
- Williams, H.P. 1974. "Experiments in the Formulation Of Integer Programming Problems." Mathematical Programming Study. 2(1974).
- \_\_\_\_\_. "Logical Problems and Integer Programming." Bulletin of the Institute of Applied Mathematics. 13(1977).
- \_\_\_\_\_. Model Building in Mathematical Programming. New York: John Wiley & Sons, 1978a.
- \_\_\_\_\_. "The Reformulation of Two Mixed Integer Programming Models." Mathematical Programming. 14(1978b):325-31.
- \_\_\_\_\_. "The Economic Interpretation of Duality for Practical Mixed Integer Programming Problems." Survey of Math Programming Society. 2(1980):567-90.
- Woolsey, R.E.D. "How to Do Integer Programming in the Real World." in Integer Programming, H.M. Salkin, (ed.), Addison-Wesley, Chapter 13, 1975.

- Wyman, F.P. "Binary Programming: A Decision Rule for Selecting Optimal vs. Heuristic Techniques." The Computer Journal. 16(1973):135-40.
- Zanakis, S.H. "Heuristic 0-1 Linear Programming: An Experimental Comparison of Three Methods." Management Science. 24(1977):91-104.
- Zanakis, S.H. and J.R. Evans. "Heuristic Optimization: Why, When and How to Use it." Interfaces. 11(1981):84-90.
- Zionts, S. Linear and Integer Programming. Prentice Hall, Englewood Cliffs, NJ, 1973.

**Table 15.1.**


---

Maximize	$7X_1$	$-3X_2$	$-10X_3$	
	$X_1$	$-2X_2$		# 0
	$X_1$		$-20X_3$	# 0
	$X_1 \leq 0$	$X_2 \leq 0$ integer	$X_3, 0,1$	

---

**Table 15.2. GAMS Input for Example Integer Program**


---

```

5  POSITIVE VARIABLE      X1
6  INTEGER VARIABLE       X2
7  BINARY VARIABLE        X3
8  VARIABLE                OBJ
9
10 EQUATIONS               OBJF
11                          X1 X2
12                          X1 X3;
13
14 OBJF..      7*X1-3*X2-10*X3 =E= OBJ;
15 X1X2..      X1-2*X2 =L=0;
16 X1X3..      X1-20*X3 =L=0;
17
18 MODEL IPTEST /ALL/;
19 SOLVE IPTEST USING MIP MAXIMIZING OBJ;

```

---

Figure 15.1 Graph of Feasible Integer Points for First LP Problem

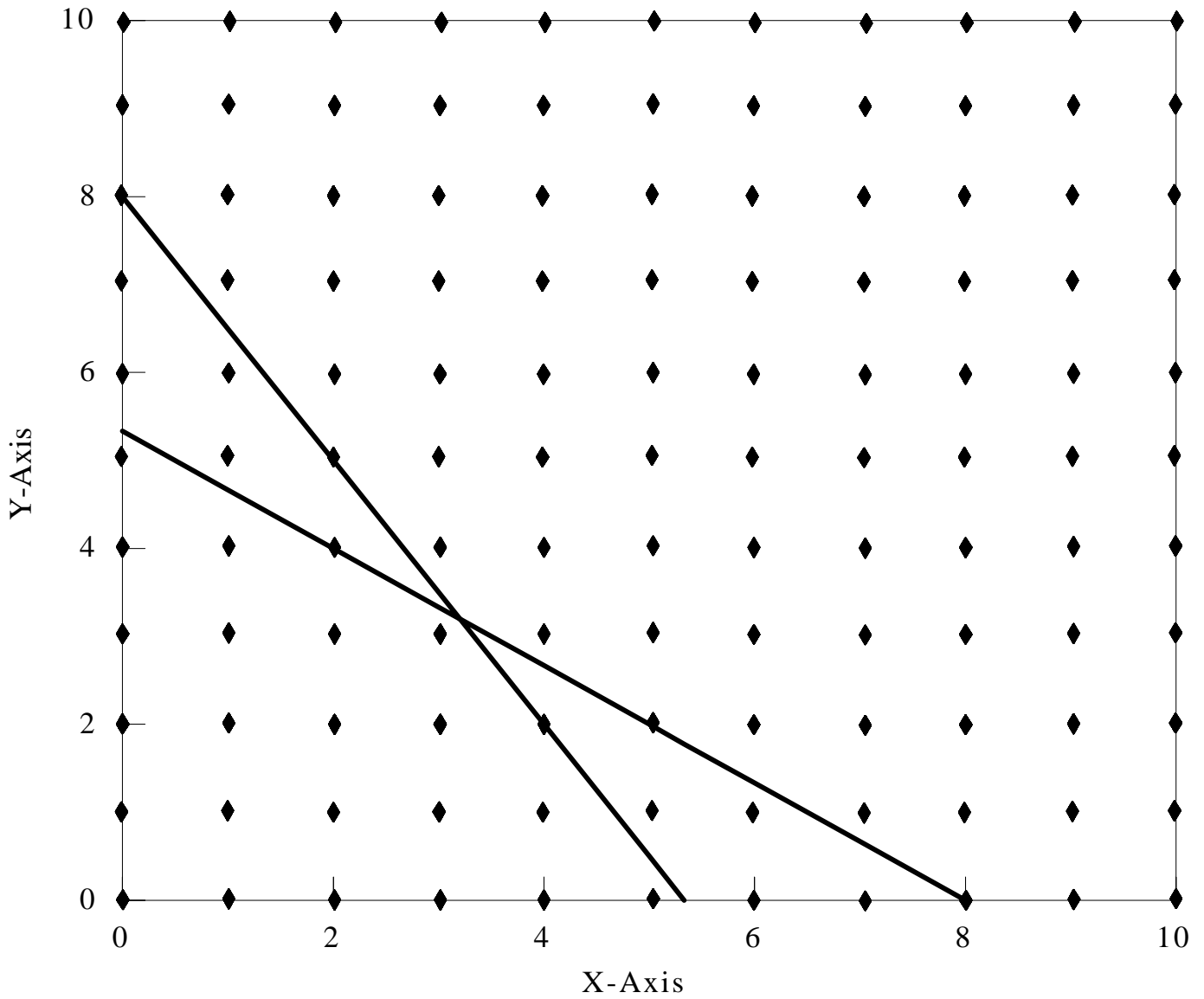
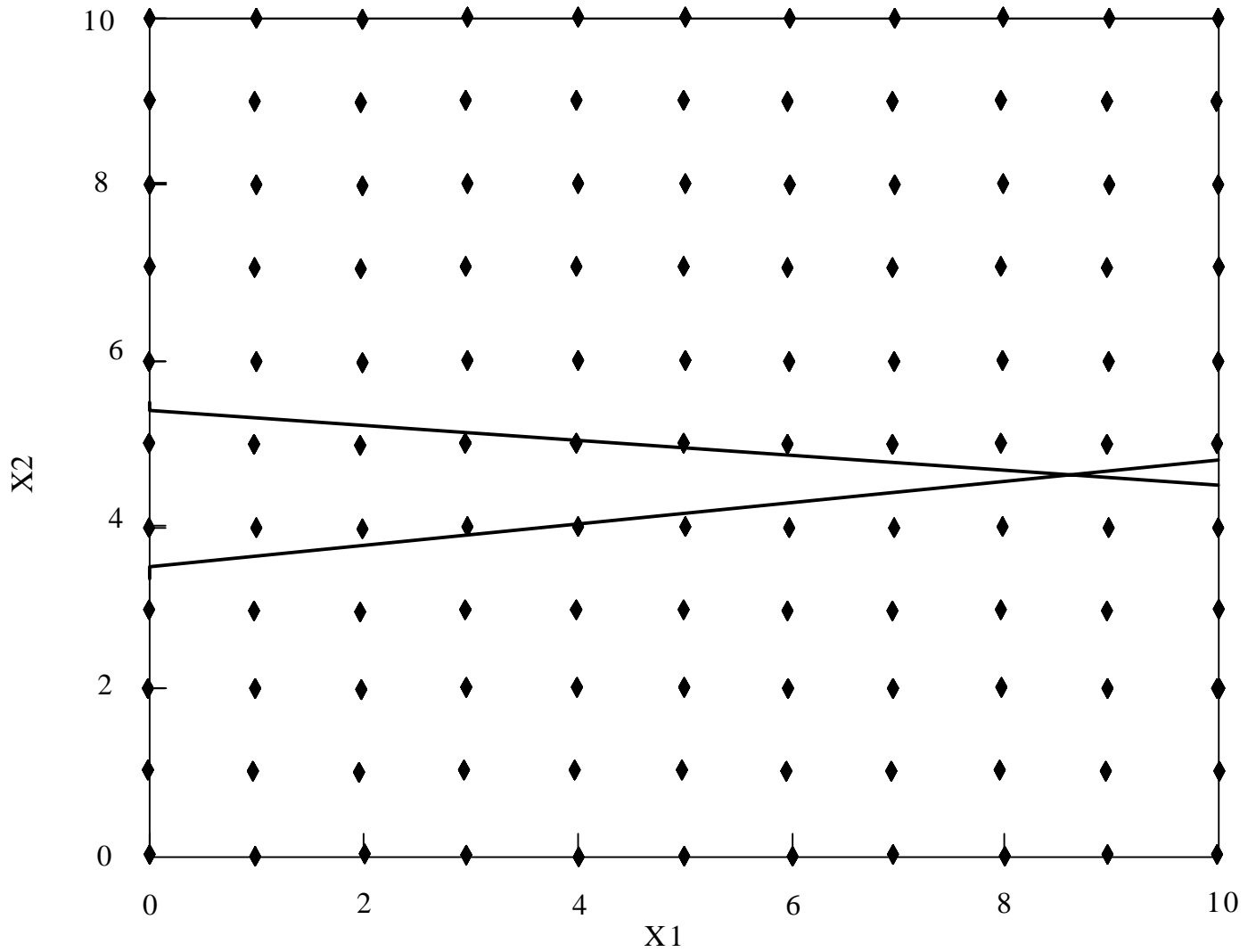
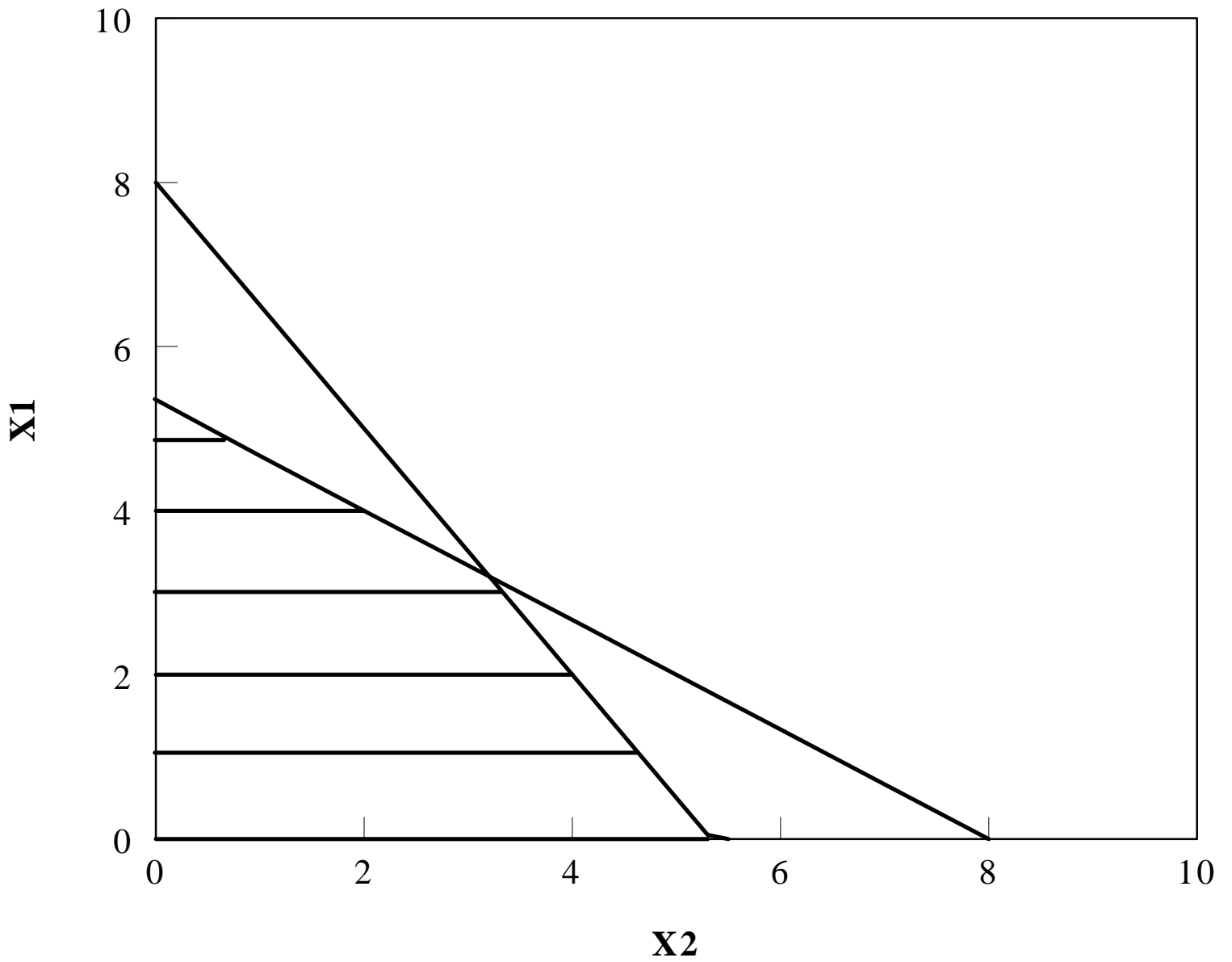


Figure 15.2 Graph of Feasible Integer Points for Second Integer Problem



**Figure 15.3 Mixed Integer Feasible Region**





## CHAPTER XVI: INTEGER PROGRAMMING FORMULATIONS

IP is a powerful technique for the formulation of a wide variety of problems. This section presents a number of common formulations.

### 16.1 Knapsack - Capital Budgeting Problem

The knapsack problem, also known as the capital budgeting or cargo loading problem, is a famous IP formulation. The knapsack context refers to a hiker selecting the most valuable items to carry, subject to a weight or capacity limit. Partial items are not allowed, thus choices are depicted by zero-one variables. The capital budgeting context involves selection of the most valuable investments from a set of available, but indivisible, investments subject to limited capital availability. The cargo loading context involves maximization of cargo value subject to hold capacity and indivisibility restrictions.

The general problem formulation assuming only one of each item is available is

$$\begin{aligned} \text{Max} \quad & \sum_j v_j X_j \\ \text{s.t.} \quad & \sum_j d_j X_j \leq W \\ & X_j = 0 \text{ or } 1 \text{ for all } j \end{aligned}$$

The decision variables indicate whether the  $j^{\text{th}}$  alternative item is chosen ( $X_j=1$ ) or not ( $X_j=0$ ). Each item is worth  $v_j$ . The objective function gives the total value of all items chosen. The capacity used by each  $X_j$  is  $d_j$ . The constraint requires total capacity use to be less than or equal to the capacity limit ( $W$ ).

#### 16.1.1 Example

Suppose an individual is preparing to move. Assume a truck is available that can hold at most 250 cubic feet of items. Suppose there are 10 items which can be taken and that their names, volumes and values are as shown in Table 16.1. The resultant formulation is

$$\begin{aligned} \text{Max} \quad & 17x_1 + 5x_2 + 22x_3 + 12x_4 + 25x_5 + x_6 + 15x_7 + 21x_8 + 5x_9 + 20x_{10} \\ \text{s.t.} \quad & 70x_1 + 10x_2 + 20x_3 + 20x_4 + 15x_5 + 5x_6 + 120x_7 + 5x_8 + 20x_9 + 20x_{10} \leq 250 \\ & x_j = 0 \text{ or } 1, \text{ for all } j \end{aligned}$$

The GAMS formulation is called KNAPSACK. The optimal objective function value equals 128. The values of the variables and their respective reduced costs are shown in Table 16.2. This solution indicates that all items except furniture,  $X_7$ , should be taken.

There are a couple of peculiarities in this solution which should be noted. First, the constraint has 65 units in slack (250-185) and no shadow price. However, for practical purposes the constraint does have a shadow price as the  $X_7$  variable would come into the solution if there were 120 more units of capacity, but slack is only 65. Further, note that each of the variables has a non-zero reduced cost. This is because this particular problem was solved with the GAMS version of OSL, a branch and bound type algorithm

and each of these variables was bounded at one. Thus, they have reduced costs reflecting bounds requiring the variables to equal either zero or one. These data are misleading as indicated in the discussion in the previous chapter on IP shadow prices.

### **16.1.2 Comments**

The knapsack problem has been the subject of considerable theoretical interest and several applications (see von Randow; Salkin, 1975a). Armstrong, Sinha, and Zoltners provide a recent application. The capital budgeting problem context has been extensively studied (Weingartner 1963, 1966, von Randow). Variants include the cutting stock problem, where one explores the best way to cut up items such as logs, sheets of veneer, and plywood, (Eisemann and Golden). Knapsack problems also commonly appear as subproblems in algorithmic approaches to problems as shown by Williams (1978a) and Geoffrion and McBride.

The knapsack formulation contains a number of simplifying assumptions. First, the formulation permits no more than one unit of any item. This assumption could be relaxed by changing from zero-one to integer variables with constraints on item availability. Second, the value and resource usage of the items are assumed independent of the mix of items chosen. However, there may be interactions where the value of the one item is increased or decreased when certain other items are also chosen. Thus, one might need to include formulation features involving multiplication of zero-one variables. Third, capacity available is assumed independent of the value of the resource. One could relax this assumption and put in a supply curve representation.

## **16.2 Warehouse Location**

Warehouse location problems are commonly formulated as integer programs. They involve location of warehouses within a transportation system so as to minimize overall costs. The basic decision involves tradeoffs between fixed warehouse construction costs and transportation costs. In agriculture, this formulation has been used in the location of high volume grain handling facilities (Hilger, McCarl and Uhrig) and agricultural processing facilities (Fuller, Randolph and Klingman; Faminow and Sarhan). The plant, store and distribution center location problems are closely related (von Randow). A general warehouse location problem formulation is as follows:

$$\begin{array}{ll}
\text{Min} & \sum_k F_k V_k \quad + \quad \sum_{i,k} C_{ik} X_{ik} \quad + \quad \sum_{k,j} D_{kj} Y_{kj} \quad + \quad \sum_{i,j} E_{ij} Z_{ij} \\
\text{s.t.} & \sum_k X_{ik} \leq S_i \quad \text{for all } i \\
& \sum_k Y_{kj} \geq D_j \quad \text{for all } j \\
& \sum_i X_{ik} \leq Y_{kj} \quad \text{for all } k \\
& \sum_j Y_{kj} \leq CAP_k V_k \quad \text{for all } k \\
& \sum_k A_{mk} V_k \leq b_m \quad \text{for all } m \\
& V_k \in \{0, 1\}, \quad X_{ik}, Y_{kj}, Z_{ij} \geq 0 \quad \text{for all } i, j, k
\end{array}$$

This is an extension of the basic transportation problem containing intermediate shipments (transshipments) into warehouses from supply points ( $X_{ik}$ ) and from warehouses to demand points ( $Y_{kj}$ ). The formulation also contains fixed cost and new warehouse capacity considerations. The variables in the formulation are:

- $V_k$  - a zero-one indicator variable indicating whether the  $k^{\text{th}}$  warehouse is constructed;
- $X_{ik}$  - a continuous variable indicating the quantity shipped from supply point  $i$  to warehouse  $k$ ;
- $Y_{kj}$  - a continuous variable indicating the quantity shipped from warehouse  $k$  to demand point  $j$ ;
- $Z_{ij}$  - a continuous variable indicating the quantity shipped from supply point  $i$  directly to demand point  $j$ .

The problem is also characterized by a number of parameters.  $F_k$  - the fixed cost associated with construction of the  $k^{\text{th}}$  warehouse. This cost should be developed so that it represents the cost incurred during the period of time represented by the supply and demand constraints;  $CAP_k$  - the capacity of the  $k^{\text{th}}$  warehouse during the time frame leading to the supply and demand quantities;  $A_{mk}$  - the amount of the  $m^{\text{th}}$  configuration constraint used when constructing the  $k^{\text{th}}$  warehouse;  $C_{ik}$  - the cost of shipping from supply point  $i$  to warehouse  $k$ ;  $D_{kj}$  - the cost of shipping from warehouse  $k$  to demand point  $j$ ;  $E_{ij}$  - the cost of shipping from supply point  $i$  to demand point  $j$ ;  $D_j$  - the amount of demand which must be filled at the  $j^{\text{th}}$  demand point in the time period modeled;  $S_i$  - the amount of supply available at  $i^{\text{th}}$  supply point in the time period modeled;  $b_m$  - the upper limit on the  $m^{\text{th}}$  configuration constraint.

The objective function depicts total cost minimization where total cost includes warehouse construction plus shipping costs for shipments a) to warehouses, b) from warehouses, and c) directly to final demand points. The first constraint equation balances outgoing shipments with available supply for a supply point. The second constraint gives the demand requirements by demand location and requires a minimum level of incoming shipments from warehouses and supply locations. The third constraint requires outgoing shipments at a warehouse location not to exceed incoming shipments to that warehouse. The next constraints both involve our zero-one warehouse variables imposing prospective warehouse capacity using the modeling approach in the fixed cost discussion in chapter 15. Outgoing shipments are balanced with

constructed warehouse capacity. When the warehouse is not constructed then outgoing shipments must equal zero. Thus, warehouses can only be used when constructed. The last constraint limits warehouse construction through configuration constraints. Many different types of constraints could be included here, dependent on the problem setting. An example is given below.

### **16.2.1 Example**

Suppose a firm can construct a warehouse at one of three sites (A,B,C). Currently, the firm has two supply points and ships to two demand points with annual demand requirements and supply capacity given in Table 16.3. Further suppose that the potential warehouses have annual capacity and fixed cost as shown in Table 16.4. If warehouse B were constructed its annual capacity would be 60, it would cost \$720 for the 12 year life or, assuming straight line depreciation, \$60 per year. Suppose that the firm has developed a transport cost matrix as shown in Table 16.5. Finally suppose only one warehouse can be built.

This leads to the formulation shown in Table 16.6. The objective function minimizes the annual fixed cost of warehouses plus the annual variable cost of shipping. The constraints impose maximum supply constraints at two locations, minimum demand constraints at two locations, supply/demand balances at three warehouses, balances between capacity and warehouse use at three warehouses, and a constraint that requires only one of the three warehouses be constructed (i.e., a configuration constraint). Warehouse 1 capacity is set to 9999 which effectively makes its capacity unlimited if it is constructed. The GAMS formulation is called WAREHOUS.

In the solution to this model, the objective function value equals 623, and the variable and equation solutions are shown in Table 16.7. This solution corresponds to the company constructing warehouse C. The shipment pattern involves shipping 70 units from supply point 2 to warehouse C, 20 units from warehouse C to demand point 1, and 50 units from C to demand point 2. In addition, 5 units are shipped directly from supply point 2 to demand point 1 while 50 units are shipped from supply point 1 to demand point 1. The shadow prices reflect demand at point 1 costing 7 units on the margin and a cost of 5 units at demand point 2. Additional supply is worth \$3 a unit at the first supply point and \$0 a unit at the second supply point.

### **16.2.2 Comments**

This formulation is simplified. One could have a number of complications such as cost-volume relationships, or multiple warehouse alternatives at a site. Those interested in related work and extensions should see the papers by Geoffrion (1975); Francis and Goldstein; Francis, McGinnis, and White; McGinnis; Fuller, Randolph, and Klingman; Hilger, McCarl, and Uhrig; or Geoffrion and Graves.

## **16.3 Traveling Salesman Problem**

Another common IP formulation is the "Traveling Salesman Problem" (Burkard; Bellmore and Nemhauser). This problem involves developing a minimum cost route for a salesman visiting N cities then returning home. The basic problem involves selection of a route visiting all cities which minimizes the total travel cost. The machine shop scheduling may also be formulated as a travelling salesman problem (Pickard and Queyranne).

The basic problem formulation is much like the assignment problem and is:

$$\begin{aligned}
 \text{Min} \quad & \sum_{i=1}^n \sum_{j=1}^n d_{ij} X_{ij} \\
 \text{s.t.} \quad & \sum_{j=1}^n X_{ij} = 1 \quad \text{for all } i \\
 & \sum_{i=1}^n X_{ij} = 1 \quad \text{for all } j \\
 & X_{ij} = 0 \text{ or } 1 \quad \text{for all } i \text{ and } j \text{ where } i \neq j
 \end{aligned}$$

The decision variable ( $X_{ij}$ ) equals one if the salesman goes from city  $i$  to city  $j$ , and zero otherwise. The possibility of moving from any city to itself is precluded. There is a known cost of moving from city  $i$  to city  $j$  ( $d_{ij}$ ). The objective function gives the total cost of completing the route which will be minimized. The first constraint states that the salesman must leave each city once. The second constraint states that the salesman must enter each city once. All decision variables are restricted to equal either zero or one.

The above formulation is that of the classical assignment problem (Wagner); however, it is not yet a complete traveling salesman formulation. There is a difficulty that often arises, known as a subtour. Consider a 5-city problem in which the optimum solution consists of  $X_{12}=1$ ,  $X_{23}=1$ ,  $X_{31}=1$ ,  $X_{45}=1$  and  $X_{54}=1$ . This solution is feasible in the above formulation and could be minimum distance. However, it reflects a disjointed trip in which one salesman goes from city 1 to city 2 to city 3 and back to city 1 without visiting cities 4 and 5, while another salesman goes from city 4 to city 5 and back to city 4. This solution exhibits so-called subtours, disjoint loops of a size less than the number of cities. Such subtours can be of any size involving two, three, four, or any number of cities up to the number in the problem minus two, although empirical evidence (cited in Garfinkel and Nemhauser; Bellmore and Nemhauser) indicates that subtours of more than four or five cities do not appear in practice. The prohibition of subtours requires additional constraints. The subtours could be eliminated by the imposition of the following constraints:

Two City

$$X_{ij} + X_{ji} \leq 1 \quad \text{for all } i \text{ and } j \quad \text{where } i \neq j$$

Three City

$$X_{ij} + X_{jk} + X_{ki} \leq 2 \quad \text{for all } i, j \text{ and } k \quad \text{where } i \neq j \neq k$$

Four City

$$X_{ij} + X_{jk} + X_{kL} + X_{Li} \leq 3 \quad \text{for all } i, j, k, \text{ and } L \quad \text{where } i \neq j \neq k \neq L$$

The first set of constraints renders all two-city subtours infeasible enforcing mutual exclusivity between the variables representing travel from city i to city j and travel from city j to city i. The next constraint set precludes three city subtours prohibiting travel from i to j then on to k, finally from k back to i. Here only two of the three activities are allowed in the solution. Similarly, the four-city subtour constraints prevent one from traveling from city i to city j, then j to k, and on from k to L, and from L back to i.

In a practical problem this way of dealing with subtours would produce a very large constraint set. For example, with 30 cities there would be 870 constraints for the prevention of the two city subtours alone. In general, constraints would be required precluding subtours from size 2 up through the greatest integer number not exceeding half the number of cities. Other formulations exist which preclude subtours in a more compact fashion. Miller, Tucker, and Zemlin show that the following constraints eliminate subtours in an N city problem,

$$U_i + U_j - NX_{ij} \leq N - 1 \quad i' 2, \dots, N; \quad j' 2, \dots, N; \quad i \neq j$$

$$U_i \geq 0 \quad i' 2, \dots, N$$

where new continuous variables (U) are introduced. Dantzig, Fulkerson, and Johnson (1954) give yet another method.

**16.3.1 Example**

Consider a salesman that has to visit six cities. Suppose these cities are separated by the distances in Table 16.8 and the salesman wants to minimize total distance traveled. The example formulation appears in Table 16.9. The objective function minimizes the sum of the distance times zero-one variables indicating whether the salesman travels between cities i and j,  $X_{ij}$ . The first six constraints require that each city be left and the next six constraints require that each city be visited. Subtours are prevented by the last 20 constraints following Miller, Tucker, and Zemlin (containing the 6s in the matrix and 5s on the right-hand sides). The GAMS formulation is called TRAVEL. The solution to this problem is shown in Table 16.10.

This solution reflects the traveling salesman traveling 46 miles going from city 1 to city 2, to city 3, to city 6, to city 5, to city 4 and back to city 1, completing a loop. Subtours are not present.

### 16.3.2 Comments

This problem has been extensively studied (see reviews by Bellmore and Nemhauser; Golden and Assad; Laporte and Lawler et. al.). Unfortunately, solving this problem is very difficult because of the number of possible feasible solutions (e.g., in the six-city problem there are five factorial possible solutions). Several heuristics have been developed for this problem. It is not recommended that it be directly solved with an IP algorithm, rather heuristics are usually used. A variant of this problem involves scheduling problems (Eilon).

### 16.4 Decreasing Costs

Models may need to depict situations where volume increases lead to either marginal cost decreases or marginal revenue increases. For example such situations would occur when: a) the purchase of transportation services involves volume discounts, or b) production exhibits positive economies of scale when cost drops as more units are produced. LP cannot satisfactorily model these situations. A separable LP formulation would use the cheapest cost activity first ignoring the volume requirements necessary to incur such a cost (i.e., using the activity with lowest transportation cost at less than the required volume rather than using more expensive transport rate relevant at that lower volume). Thus, another modeling approach is required. One could use the nonlinear form of separable programming, but this would yield local optimal solutions. Alternatively, a mixed IP formulation can be used. This will be explained herein.

The basic problem in matrix form is

$$\begin{array}{ll}
 \text{Max} & eY \quad \& \quad f(Z) \\
 \text{s.t.} & Y \quad \& \quad \sum_m G_m X_m \quad \# \quad 0 \\
 & & \sum_m A_m X_m \quad \& \quad Z \quad \# \quad 0 \\
 & & \sum_m H_{im} X_m \quad \# \quad b_i \quad \text{for all } i \\
 & Y, \quad X_m, \quad Z \quad \geq \quad 0
 \end{array}$$

where Z is the quantity of input used, f(Z) is the total cost of acquiring the input which exhibits diminishing marginal cost (i.e., the per unit cost of Z falls as more is purchased); e is the sale price for a unit of output (Y);  $G_m$  is the quantity of output produced per unit of production activity  $X_m$ ;  $A_m$  is the amount of the resource which is used per unit of  $X_m$ ; and  $H_{im}$  is the number of units of the  $i^{\text{th}}$  fixed resource which is used per unit of  $X_m$ .

In this problem the objective function maximizes total revenue from product sale ( $eY$ ) less total costs ( $f(Z)$ ). The first constraint balances products sold (Y) with production ( $\sum G_m X_m$ ). The second constraint balances input usage ( $\sum A_m X_m$ ) with supply (Z). The third constraint balances resource usage by production ( $\sum H_{im} X_m$ ) with exogenous supply ( $b_i$ ). This problem may be reformulated as an IP problem

by following an approximation point approach.

$$\begin{aligned}
 \text{Max } eY & \quad \& \quad f'(Z_k^*) & \quad R_k \\
 \text{s.t. } Y & \quad \& \quad \sum_m G_m X_m & \quad \# \quad 0 \\
 & \quad \& \quad \sum_m A_m X_m & \quad \& \quad \sum_k R_k & \quad \# \quad 0 \\
 & \quad \& \quad \sum_m H_{im} X_m & \quad \# \quad b_i & \quad \text{for all } i \\
 & \quad \& \quad R_k & \quad \& \quad (Z_k^* - Z_{k-1}^*) D_k & \quad \# \quad 0 & \quad \text{for all } k \\
 & \quad \& \quad R_k & \quad \leq & \quad (Z_k^* - Z_{k-1}^*) D_{k+1} & \quad \# \quad 0 & \quad \text{for all } k \text{ but the last one} \\
 Y, & \quad X_m, & R_k & \quad \leq & 0 & \quad \text{for all } m \text{ and } k \\
 & & & & D_k & \quad \in \quad \{0, 1\} & \quad \text{for all } k
 \end{aligned}$$

The variables are  $Y$  and  $X_m$ , as above, but the  $Z$  variable has been replaced with two sets of variables:  $R_k$  and  $D_k$ . The variables  $R_k$  which are the number of units purchased at cost  $f'(Z_k^*)$ ;  $Z_k^*$  are a set of approximation points for  $Z$  where  $Z_0^* = 0$ ; where  $f'(Z_k^*)$  is the first derivative of the  $f(Z)$  function evaluated at the approximation point  $Z_k^*$ . While simultaneously the data for  $D_k$  is a zero-one indicator variable indicating whether the  $k^{\text{th}}$  step has been fully used.

The formulation insures that the proper total cost is incurred, and that the decreasing per unit costs are only used when the proper quantities are purchased. The last two constraints enforce this restriction, requiring  $R_k$  to equal  $Z_k - Z_{k-1}$  before  $R_{k+1}$  can be non-zero (i.e., the  $k^{\text{th}}$  increment must be paid for before the  $k+1^{\text{st}}$  increment can be purchased). The first three equations are as defined above. Notice that the  $k^{\text{th}}$  step variable can be no larger than  $D_k$  times the difference between  $Z_k$  and  $Z_{k-1}$ . Thus,  $R_k$  is prevented from being non-zero unless the indicator variable  $D_k$  is also non-zero. However, the last constraint imposes a relationship between the  $k^{\text{th}}$  step variable and the indicator variable for step  $k+1$ . Consequently,  $R_k$  must equal its maximum value  $(Z_{k+1} - Z_k)$  if the  $k+1^{\text{st}}$  indicator is non-zero. Similarly,  $R_1$  through  $R_{k-1}$  must equal their upper limits in order that  $R_k$  can be non-zero. Consequently, this only permits input purchases at the lower cost exhibited under the higher volumes, only if inputs have been purchased at all volumes previous to those.

### 16.4.1 Example

Consider a problem in which total cost of the input  $Z$  and the production relationships are given by

$$\begin{aligned}
 \text{Max } & 4Y & \quad \& \quad (3 + .125Z) Z \\
 & Y & \quad \& \quad 2X & \quad \# \quad 0 \\
 & & \quad \& \quad X & \quad \& \quad Z & \quad \# \quad 0 \\
 & & & \quad X & \quad \# \quad 5 \\
 Y, & X, & Z & \quad \leq & 0
 \end{aligned}$$

Suppose we approximate  $Z$  at 2, 4, 6, 8 and 10. The formulation becomes





annual costs of machinery purchase with the extra profits obtained by having that machinery. A general formulation of this problem is

$$\begin{aligned}
 \text{Max} \quad & \sum_k F_k Y_k - \sum_{j,m} C_{jm} X_{jm} \\
 \text{s.t.} \quad & \sum_{j,m} \text{Cap}_{ik} Y_k - \sum_{j,m} A_{ijkm} X_{jm} \leq 0 \quad \text{for all } i \text{ and } k \\
 & \sum_{j,m} D_{njm} X_{jm} \leq b_n \quad \text{for all } n \\
 & \sum_k G_{rk} Y_k \leq e_r \quad \text{for all } r \\
 & Y_k \text{ is a nonnegative integer, } X_{jm} \geq 0 \text{ for all } j, k, \text{ and } m
 \end{aligned}$$

The decision variables are  $Y_k$ , the integer number of units of the  $k^{\text{th}}$  type machinery purchased;  $X_{jm}$ , the quantity of the  $j^{\text{th}}$  activity produced using the  $m^{\text{th}}$  machinery alternative. The parameters of the model are:  $F_k$ , the annualized fixed cost of the  $k^{\text{th}}$  machinery type;  $\text{Cap}_{ik}$ , the annual capacity of the  $k^{\text{th}}$  machinery type to supply the  $i^{\text{th}}$  resource;  $G_{rk}$ , the usage of the  $r^{\text{th}}$  machinery restriction when purchasing the  $k^{\text{th}}$  machinery type;  $C_{jm}$ , the per unit net profit of  $X_{jm}$ ;  $A_{ijkm}$ , the per unit use by  $X_{jm}$  of the  $i^{\text{th}}$  capacity resource supplied by purchasing machine  $k$ ;  $D_{njm}$ , the per unit usage of fixed resources of the  $n^{\text{th}}$  type by  $X_{jm}$ ;  $b_n$ , the endowment of the  $n^{\text{th}}$  resource in the year being modeled; and  $e_r$ , the endowment of the  $r^{\text{th}}$  machinery restriction.

The objective function maximizes profits from machinery operation less the fixed costs of acquisition. The first constraint balances the capacity of the machinery purchased with the use of that capacity. These constraints preclude machinery from being used unless it is purchased. The second constraint imposes constraints on resources other than machinery. The third constraint imposes configuration constraints on machinery purchases.

### 16.5.1 Example

Assume that a farm is considering the purchase of equipment involving a choice of two tractors, two plows, two discs, two planters and two harvesting units. The working rates and costs are given in Table 16.12. Time available by period is given in Table 16.13. The farm has 600 acres. Machinery resource calculations are shown in Table 16.14. Yields, prices, and costs are given in Table 16.15.

Three operations are done on the farm: plowing, simultaneous discing and planting, and harvesting; plowing is done in time periods 1-2; disc-planting in period 2 and harvesting in period 3. In addition, when buying the equipment, one must match the disc and the planter; disc number one can be purchased only with planter number one and disc number two only with planter number two. The formulation is given in Table 16.16 and in file MACHSEL. The solution to this IP problem yields an IP objective function of 116,100 when it is solved as an LP its objective function equals 124,301. The values of the solution variables are

given in Table 16.17.

### **16.5.2 Comments**

This formulation has been used in agricultural economics. For example see the machinery selection work by Danok, McCarl, and White (1978, 1980); Clayton and McCarl; or Baker, Dixit, and McCarl.

### **16.6 Other Formulations**

While several classes of formulations were addressed above, there are numerous other formulations which could have been included. Here we mention networks, dynamic programming, scheduling, and combinatorial problems.

The vast majority of network problems are integer by nature. Many of them yield integer solutions because of the structure of the basis (Wagner, 1969). These types of problems are the assignment, transportation, transshipment, shortest path, maximal flow, and minimum spanning tree. A general presentation can be seen in Kennington and Helgeson; Bazaraa, Jarvis and Sherali; or Jensen and Barnes.

A second related class of problems are dynamic programming problems. Many dynamic programming algorithms involve integer valued variables. Many common IP problems have been cast as dynamic programming problems; e.g., Nemhauser mentions network, traveling salesmen and scheduling problems as places where dynamic programming has been applied.

There is also a large class of integer scheduling problems. One such problem is the vehicle scheduling problem where buses, aircraft, or ships are routed to places where items need to be delivered. Wagner (1969), and Markowitz and Manne give early developments and references to solve this class of problems. While Assad and Golden give more recent references there have been a vast number of machine scheduling applications involving assembly line balancing, flow shop scheduling, batch sizing, etc. Eilon reviews this topic, and von Randow gives 13 pages of references. Project scheduling problems have also been formulated (Davis, Patterson).

Another class of integer problems are the combinatorial problems, most of which can be formulated as IP problems. These include network type problems such as maximum flow problems, set covering, matching problems, weighted matching problems, spanning trees, and traveling salesmen problems. Many of these problems are classed as very difficult to solve. The book by Papadimitriou and Steiglitz gives background and formulations.

Finally, we should mention that new applications of IP are developed virtually every day. For example, von Randow, in a bibliography of studies between 1978 and 1981, gives 130 pages of citations to IP relating mainly to that time period. Thus, there are many classes of problems that we have not covered above.

## References

- Armstrong, R.D., P. Sinha and A.A. Zoltners. "The Multiple Choice Nested Knapsack Model." Management Science. 28(1982):34-43.
- Baker, T.G., P. Dixit, and B.A. McCarl. Documentation and Use of a Computer Software Package for Simulation of Optimal Investment in Dairy Processing. Purdue University Agricultural Experiment Station; Bulletin. 345(1981).
- Bazaraa, M.S., J. Jarvis, and H. Sherali. Linear Programming and Network Flows. John Wiley & Sons, 1977.
- Bellmore, M. and G. Nemhauser. "The Traveling Salesman Problem: A Survey." Operations Research. 16(1968):538-58.
- Burkard, R.E. "Travelling Salesman and Assignment Problems: A Survey." in P.L. Hammer, E.L. Johnson, B.H. Korte (eds.) "Discrete Optimization 1" Annals of Discrete Mathematics 4. Amsterdam, North Holland, 1979:193-215.
- Clayton, K. and B.A. McCarl. "Management of Solid Waste in Systems Including Nonmetropolitan Areas, with Emphasis on Resource Recovery." North Central Journal of Agricultural Economics. 1(1979):61-72.
- Danok, A.B., B.A. McCarl, and T.K. White. "Machinery Selection and Crop Planning on a State Farm in Iraq." American Journal of Agricultural Economics. 60(1978):544-549.
- Danok, A.B., B.A. McCarl, and T.K. White. "Machinery Selection Modeling: Incorporation of Weather Variability." American Journal of Agricultural Economics. 62(1980):700-708.
- Dantzig, G.B., D.R. Fulkerson, and S.M. Johnson. "Solution of a Large Scale Travelling Salesman Problem." Operations Research 2(1954):393-410.
- Davis, E.W. "Project Scheduling Under Resource Constraints: Historical Review and Categorization of Procedures." American Institute of Industrial Engineers Transactions. 5(1973):297-313.
- Eisemann, K. "The Trim Problem." Management Science. 3(1957):279-284.
- Eilon, S. "More Against Optimization." Omega. 5(1977):627-633.
- Faminow, M.D. and M.E. Sarhan. "The Location of Fed Cattle Slaughtering and Processing: An Application of Mixed-Integer Programming." Canadian Journal of Agricultural Economics. (1983):425-436.
- Francis, R. and J. Goldstein. "Location Theory: A Selective Bibliography." Operations Research. 22(1974):400-409.
- Francis, R.L., L.F. McGinnis and J.A. White. "Locational Analysis." European Journal of Operational Research. 12(1983):220-252.
- Fuller, W.W., P. Randolph, and D. Klingman. "Optimizing Subindustry Marketing Organizations: A Networked Analysis Approach." American Journal of Agricultural Economics. 58(1976):425-436.
- Garfinkel, T.S. and G.L. Nemhauser. Integer Programming. New York: John Wiley and Sons, (1971).

- Geoffrion, A.M. "A Guide to Computer-Assisted Methods for Distribution Systems Planning." Sloan Management Review. 16(1975):17-41.
- Geoffrion, A.M. and G.W. Graves. "Multicommodity Distribution System Design by Bender's Decomposition." Management Science. 20(1974):822-844.
- Geoffrion, A.M. and R. McBride. "Lagrangian Relaxation Applied to Capacitated Facility Location Problems." American Institute of Industrial Engineers Transactions. 10(1978):40-47.
- Golden, B. "Approaches to the Cutting Stock Problem." American Institute of Industrial Engineers Transactions. 8(1976):265-274.
- Hilger, B.A., B.A. McCarl, and J.W. Uhrig. "Facilities Location: the Case of Grain Subterminals." American Journal of Agricultural Economics. 59(1977):674-682.
- Jensen, P.A. and J.W. Barnes. Network Flow Programming. John Wiley & Sons, New York, 1980.
- Kennington, J. and R. Helgeson. Algorithms for Network Programming. John Wiley & Sons, New York, 1980.
- Markowitz, H.M. and A Manne. "On the Solution of Discrete Programming Problems." Econometrica. 25(1975):84-110.
- McGinnis, L. "A Survey of Recent Results for a Class of Facility Location Problems." American Institute of Industrial Engineers Transactions. 9(1977):11-18.
- Miller, C.E., A.W. Tucker, and R.A. Zemlin. "Integer Programming Formulation of Travelling Salesmen Problems." Journal of the Association for Computing Machinery. 7(1960):326-329.
- Nemhauser, G.L. Introduction to Dynamic Programming. New York: John Wiley & Sons, 1966.
- Papadimitriou, C.H. and K. Steiglitz. Combinatorial Optimization: Algorithms and Complexity. Prentice-Hall, Englewood Cliffs, NJ. 1982.
- Patterson, J.H. "A Comparison of Exact Approaches for Solving the Multiple Constrained Resource, Project Scheduling Problem." Management Science. 30(1984):854-867.
- Pickard, J.C. and M. Queyranne. "The Time-Dependent Travelling Salesman Problem and Its Application to the Tardiness Problem in One-Machine Scheduling." Operations Research. 2(1978):86-110.
- Salkin, H. "The Knapsack Problem." Naval Research Logistics Quarterly. 22(1975):127-155.
- Von Randow, R. Integer Programming and Related Areas-A Classified Bibliography. 1978-1981. Springer-Verlag, New York, 1982:338.
- Wagner, H.M. Principles of Operations Research: With Applications to Managerial Decisions. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1969.
- Weingartner, H.M. Mathematical Programming and the Analysis of Capital Budgeting Problems. Englewood Cliffs, NJ: Prentice-Hall, 1963.
- \_\_\_\_\_. "Capital Budgeting of Interrelated Projects, Survey and Synthesis." Management Science. 12(1966):485-516.
- Williams, H.P. Model Building in Mathematical Programming. New York: John Wiley & Sons, 1978.

**Table 16.1. Items for the Knapsack Example Problem**

Variable	Item Name	Item Volume (Cubic feet)	Item Value (\$)
X <sub>1</sub>	Bed and mattress	70	17
X <sub>2</sub>	TV set	10	5
X <sub>3</sub>	Turntable and records	20	22
X <sub>4</sub>	Armchairs	20	12
X <sub>5</sub>	Air conditioner	15	25
X <sub>6</sub>	Garden tools and fencing	5	1
X <sub>7</sub>	Furniture	120	15
X <sub>8</sub>	Books	5	21
X <sub>9</sub>	Cooking utensils	20	5
X <sub>10</sub>	Appliances	20	20

**Table 16.2. Solution to the Knapsack Example Problem**

Obj = 128			
Variable	Value	Reduced Cost	
X <sub>1</sub>	1	17	
X <sub>2</sub>	1	5	
X <sub>3</sub>	1	22	
X <sub>4</sub>	1	12	
X <sub>5</sub>	1	25	
X <sub>6</sub>	1	1	
X <sub>7</sub>	0	15	
X <sub>8</sub>	1	21	
X <sub>9</sub>	1	5	
X <sub>10</sub>	1	20	
Constraint	Activity	Shadow Price	
Space	185	0	

**Table 16.3. Supply/Demand Information for Warehouse Location Example**

Total Supply		Total Demand	
Point	Units	Point	Units
1	50	1	75
2	75	2	50

**Table 16.4. Warehouse Capacities and Costs for the Warehouse Location Example**

Warehouse	Annual Capacity	Fixed Cost/Life (\$)	1 -Year Cost
A	Unlimited	500/10 years	\$50
B	60	720/12 years	\$60
C	70	680/10 years	\$68

**Table 16.5. Transport Costs (in \$/unit) for Warehouse Location Example**

		Shipping Point				
		Supply		Warehouse		
		1	2	A	B	C
Warehouse	A	1	6	-	-	-
	B	2	3	-	-	-
	C	8	1	-	-	-
Demand	1	4	7	4	3	5
	2	8	6	6	4	3

Table 16.6. Formulation of the Warehouse Location Example Problem																			
$V_A$	$V_B$	$V_C$	$X_{1A}$	$X_{1B}$	$X_{1C}$	$X_{2A}$	$X_{2B}$	$X_{2C}$	$Y_{A1}$	$Y_{A2}$	$Y_{B1}$	$Y_{B2}$	$Y_{C1}$	$Y_{C2}$	$Z_{11}$	$Z_{12}$	$Z_{21}$	$Z_{22}$	RHS
50	60	68	1	2	8	6	3	1	4	6	3	4	5	3	4	8	7	6	Min
									1		1		1		1		1		# 75
			1	1	1					1		1		1		1		1	# 50
						1	1	1								1		1	\$ 50
									1	1							1	1	\$ 75
			-1			-1			1										# 0
				-1			-1				1	1							# 0
					-1			-1					1	1					# 0
-9999									1	1									# 0
	-60										1	1							# 0
		-70											1	1					# 0
1	1	1																	# 1
			$V_A, V_B, V_C$	0	(0,1)			$X_{ik}$							$Y_{kj}$		$Z_{ij}$		\$ 0 for all i, j, k



**Table 16.7. Solution Results for the Warehouse Location Example**

---

Obj = 623

Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
V <sub>A</sub>	0	0	1	0	-3.00
V <sub>B</sub>	0	2	2	0	0
V <sub>C</sub>	1	0	3	0	7.00
X <sub>1A</sub>	0	0	4	0	5.00
X <sub>1B</sub>	0	2.00	5	0	-4
X <sub>1C</sub>	0	10.00	6	0	-3.00
X <sub>2A</sub>	0	2	7	0	-1.00
X <sub>2B</sub>	0	0	8	0	-0.05
X <sub>2C</sub>	70	0	9	0	-1.00
Y <sub>A1</sub>	0	1.052	10	0	-1.00
Y <sub>A2</sub>	0	5.052	11	0	-2
Y <sub>B1</sub>	0	0			
Y <sub>B2</sub>	0	3.00			
Y <sub>C1</sub>	20	0			
Y <sub>C2</sub>	50	0			
Z <sub>11</sub>	50	0			
Z <sub>12</sub>	0	6.00			
Z <sub>21</sub>	5	0			
Z <sub>22</sub>	0	1.00			

---

**Table 16.8. Distances Between Cities for the Travelling Salesman Problem**

	1	2	3	4	5	6
1	--	11	7	6	8	14
2	11	--	7	9	12	13
3	7	7	--	3	7	8
4	6	9	3	--	4	8
5	8	12	7	4	--	10
6	14	13	8	8	10	--



**Table 16.10. Solution to the Travelling Salesman Example**

Obj = 46						
Variable	Value	Reduced Cost	Equation	Slack	Shadow Price	
X <sub>12</sub>	1.00	11	Leave 1	0	0	
X <sub>13</sub>	0	7	Leave 2	0	0	
X <sub>14</sub>	0	6	Leave 3	0	0	
X <sub>15</sub>	0	8	Leave 4	0	0	
X <sub>16</sub>	0	14	Leave 5	0	0	
X <sub>21</sub>	0	11	Leave 6	0	0	
X <sub>23</sub>	1.00	7	Enter 1	0	0	
X <sub>24</sub>	0	9	Enter 2	0	0	
X <sub>25</sub>	0	12	Enter 3	0	0	
X <sub>26</sub>	0	13	Enter 4	0	0	
X <sub>31</sub>	0	7	Enter 5	0	0	
X <sub>32</sub>	0	7	Enter 6	0	0	
X <sub>34</sub>	0	3	Subtour 23	4	0	
X <sub>35</sub>	0	7	Subtour 24	8	0	
X <sub>36</sub>	1.00	8	Subtour 25	7	0	
X <sub>41</sub>	1.00	6	Subtour 26	0	0	
X <sub>42</sub>	0	9	Subtour 32	0	0	
X <sub>43</sub>	0	3	Subtour 34	9	0	
X <sub>45</sub>	0	4	Subtour 35	8	0	
X <sub>46</sub>	0	8	Subtour 36	7	0	
X <sub>51</sub>	0	8	Subtour 42	2	0	
X <sub>52</sub>	0	12	Subtour 43	1	0	
X <sub>53</sub>	0	7	Subtour 45	4	0	
X <sub>54</sub>	1.00	4	Subtour 46	3	0	
X <sub>56</sub>	0	10	Subtour 52	3	0	
X <sub>61</sub>	0	14	Subtour 53	2	0	
X <sub>62</sub>	0	13	Subtour 54	0	0	
X <sub>63</sub>	0	8	Subtour 56	4	0	
X <sub>64</sub>	0	8	Subtour 62	4	0	
X <sub>65</sub>	1.00	10	Subtour 63	3	0	
U <sub>2</sub>	0	0	Subtour 64	7	0	
U <sub>3</sub>	1	0	Subtour 65	0	0	
U <sub>4</sub>	4	0				
U <sub>5</sub>	3	0				
U <sub>6</sub>	2	0				

**Table 16.11. Solution to the Decreasing Costs Example**

Variable	Objective function = 29.50				
	Value	Reduced Cost	Equation	Slack	Shadow Price
Y	10	0	Y balance	0	4.0
X	5	6.5	Z balance	0	1.5
R <sub>1</sub>	2	0	R <sub>1</sub> D <sub>1</sub>	0	0
R <sub>2</sub>	2	0	R <sub>2</sub> D <sub>2</sub>	0	0
R <sub>3</sub>	1	0	R <sub>3</sub> D <sub>3</sub>	1	0
R <sub>4</sub>	0	0	R <sub>4</sub> D <sub>4</sub>	0	0.5
R <sub>5</sub>	0	0	R <sub>5</sub> D <sub>5</sub>	0	1.0
D <sub>1</sub>	1	0	R <sub>1</sub> D <sub>2</sub>	0	1.0
D <sub>2</sub>	1	-2	R <sub>2</sub> D <sub>3</sub>	0	0.5
D <sub>3</sub>	1	-1	R <sub>3</sub> D <sub>4</sub>	1	0
D <sub>4</sub>	0	1	R <sub>4</sub> D <sub>5</sub>	0	0
D <sub>5</sub>	0	2			

**Table 16.12. Data for the Machinery Selection Problem**

Equipment	Annualized Fixed Cost	Cost/Hour of Operation	Hrs. of Labor Used/Hr. of Operation	Acres Treated/Hour
Tractor 1	5,000	10.00	1.00	-
Tractor 2	9,000	10.00	1.00	-
Plow 1	1,000	2.00	0.20	5*
Plow 2	1,200	2.00	0.20	10*
Disc 1	1,000	1.20	0.10	10*
Disc 2	1,200	1.20	0.10	12**
Planter 1	2,000	3.40	0.10	--
Planter 2	2,100	3.40	0.22	--
Harvester 1	1,000	23.0	1.00	3***
Harvester 2	12,000	28.0	1.00	4***

\* Requires a tractor. Working rates are given for tractor 1; tractor 2 is twice as fast.  
 \*\* Has the same working rate as that of the disc that the planter is used with.  
 \*\*\* Uses one hour of tractor time/hour of harvesting.

**Table 16.13. Hours Available for the Machinery Selection Problem**

Time Period	Hours of Labor	Hours for Machinery
1	200	160
2	210	180
3	250	200

**Table 16.14. Machinery Usage Computations**

Operation	Tractor Used	Cost/Acre (\$)	Hrs. of Tractor/Acre	Plow Used	Hrs. Plow Use/Acre	Planter Used	Hrs. Planter Use/Acre	Disc Used	Hrs. Disc Used/Acre	Harvester Used	Hrs. Harvester Used/Acre
Plow	1	2.40	0.2	1	0.2	--	--	--	--	--	--
Plow	1	1.20	0.1	2	0.1	--	--	--	--	--	--
Plow	2	1.20	0.1	1	0.1	--	--	--	--	--	--
Plow	2	0.60	0.05	2	0.05	--	--	--	--	--	--
Plant-disc	1	1.46	0.1	--	--	1	0.1	1	0.1	--	--
Plant-disc	1	1.22	0.0833	--	--	2	0.0833	2	0.0833	--	--
Plant-disc	2	0.73	0.05	--	--	1	0.05	1	0.05	--	--
Plant-disc	2	0.61	0.04167	--	--	2	0.04167	1	0.0417	--	--
Harvest	1	11	0.333	--	--	--	--	--	--	1	0.333
Harvest	2	11	0.333	--	--	--	--	--	--	1	0.333
Harvest	1	9.5	0.25	--	--	--	--	--	--	2	0.25
Harvest	2	9.5	0.25	--	--	--	--	--	--	2	0.25

**Table 16.15. Yields, Prices, and Costs**

Non-machinery cost per acre	110
Price per unit of yield	2.5
Yield per acre	140

**Table 16.16. Formulation of the Machinery Selection Problem**

	Machinery Acquisition Integer Variables												Machinery Use Continuous Variables																							
	Tractor			Plow			Planter			Disc			Harvester			Plow with Tractor 1 and Plow 1 in Period			Plow with Tractor 2 and Plow 1 in Period			Plant Disc 8 Tractor 1 Planter			Harvest with Tractor 1 Harvester			Harvest with Tractor 2 Harvester			Input Purchases					
	1	2		1	2		1	2		1	2		1	2		1	2		1	2		1	2		1	2		1	2		1	2		1	2	
Objectives (max)	-5000	-9000	-1000	-1200	-2000	-2100	-1000	-1200	-1000	-1000	-1200	-1000	-12000	-10000	-12000	-2.4	-2.4	-1.2	-1.2	-1.2	-0.6	-1.46	-1.22	-0.73	-0.61	-0.61	-0.61	-9.33	-8.35	-9.33	-8.25	-8.25	-8.25	-110	-110	-110
Tractor 1 Capacity in Period	1															.2	.1					.1	.0833													
Tractor 2 Capacity in Period		1																																		
Plow 1 Capacity in Period			1													.1	.05																			
Plow 2 Capacity in Period				1																																
Capacity of Planter																																				
Capacity of Disc																																				
Capacity of Harvester																																				
Labor Available in Period																.24	.12	.12	.12	.06	.06	.12	.11	.06	.055	.055	.055									
Plow-Plant Sequencing																-1	-1	-1	-1	-1	-1	1	1	1	1	1	1									
Plant-Harvest Sequencing																																				
Land Available																1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1





**Table 16.17. Solution for the Machinery Selection Problem**  
 $obj = 116,100$

Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
Buy Tractor 1	1	-5,000	Tractor 1 capacity in Period 1	100	0
Buy Tractor 2	0	0	Tractor 1 capacity in Period 2	130	0
Buy Plow 1	0	0	Tractor 1 capacity in Period 3	50	0
Buy Plow 2	1	-1,200	Tractor 2 capacity in Period 1	0	12
Buy Planter 1	0	0	Tractor 2 capacity in Period 2	0	14.6
Buy Planter 2	1	-3300	Tractor 2 capacity in Period 3	0	22.26
Buy Disc 1	0	0	Plow 1 capacity in Period 1	0	6.25
Buy Disc 2	1	0	Plow 1 capacity in Period 2	0	0
Buy Harvester 1	0	0	Plow 2 capacity in Period 1	100	0
Buy Harvester 2	1	0	Plow 2 capacity in Period 2	180	0
Plow with Tractor 1 and Plow 1 in Period 1	0	-2.45	Planter 1 capacity	0	0
Plow with Tractor 1 and Plow 1 in Period 2	0	-1.20	Planter 2 capacity	130	0
Plow with Tractor 1 and Plow 2 in Period 1	600	0	Disc 1	0	0
Plow with Tractor 1 and Plow 2 in Period 2	0	0	Disc 2	130	0
Plow with Tractor 2 and Plow 1 in Period 1	0	-1.825	Harvester 1	0	50
Plow with Tractor 2 and Plow 1 in Period 2	0	-1.46	Harvester 2	50	0
Plow with Tractor 2 and Plow 2 in Period 1	0	0	Labor available in Period 1	128	0
Plow with Tractor 2 and Plow 2 in Period 2	0	0.13	Labor available in Period 2	144	0
Plant with Tractor 1 and Planter 1	0	-1.91	Labor available in Period 3	25	0
Plant with Tractor 1 and Planter 2	600	0	Plow Plant	0	230.533
Plant with Tractor 2 and Planter 1	0	-1.077	Plant Harvester	0	341.75
Plant with Tractor 2 and Planter 2	0	0	Land	0	229.333
Harvest with Tractor 1 and Harvester 1	0	-17.75	One Planter	0	0
Harvest with Tractor 1 and Harvester 2	600	0	One Disc	0	0
Harvest with Tractor 2 and Harvester 1	0	-25.17	Planter 1 to Disc 1	0	0
Harvest with Tractor 2 and Harvester 2	0	-5.565	Planter 2 to Disc 2	0	0
Sell Crop	84,000	0	Yield Balance	0	2.5
Purchase Inputs	600	0	Input Balance	0	110

## CHAPTER XVII: FIXING IMPROPERLY WORKING MODELS

Empirical models do not always yield acceptable solutions. This chapter contains discussion of unacceptable solution conditions and techniques for diagnosing the causes of such conditions.

### 17.1 Unacceptable Solution Conditions

Four cases of improper solutions can arise. First, a solver could fail exhibiting: a) a time, iteration, or resource limit; b) a lack of meaningful progress; or c) a report of numerical difficulties. Second, a solver may halt identifying that the problem is infeasible. Third, a solver may halt identifying that the problem is unbounded. Fourth, the solver may yield an "optimal," but unacceptable solution.

#### 17.1.1 Solver Failure -- Causes and Prevention

When solvers fail because of numerical difficulties or use an unrealistically large amount of resources to make little progress, the modeler is often in an awkward position. However, several actions may alleviate the situation.

One should first examine whether the model specification is proper. The section on structural checking below gives some techniques for examining model structure. In addition traditional input (commonly called MPS input) based solvers frequently fail because of improper coefficient location (although GAMS prevents some of these errors). In particular, errors can arise in MPS coefficient placement or item naming resulting in more than one (duplicate) coefficient being defined for a single matrix location. Given our concentration on the GAMS modeling system, procedures for finding duplicate coefficients will not be discussed. Nevertheless, this is probably the most common reason why MPS input based solvers run out of time.

The second reason for solver failure involves degeneracy induced cycling. Apparently, even the best solvers can become stuck or iterate excessively in the presence of massive degeneracy. Our experience with such cases indicates one should use an a priori degeneracy resolution scheme as discussed below. We have always observed reduced solution times with this modification.

Thirdly, a solver may fail citing numerical difficulties, an ill-conditioned basis or a lack of progress. Such events can be caused by model specification errors or more commonly poor scaling. Often one needs to rescale the model to narrow the disparity between the magnitude of the coefficients. Scaling techniques are discussed below.

All of the preventative techniques for avoiding solver failures can be used before solving a model. Modelers should check structure and consider scaling before attempting model solutions. However, degeneracy resolution should not usually be employed until a problem is identified.

#### 17.1.2 Unbounded or Infeasible Solutions

Often the applied modeler finds the solver has stopped, indicating that the model is infeasible or

unbounded. This situation, often marks the beginning of a difficult exercise directed toward finding the cause of the infeasibility or unboundedness, particularly when dealing with large models. There are several techniques one can use when this occurs. The first involves structural checking to find obvious model formulation defects. The second and third techniques involve the use of artificial variables and large upper bounds to find difficulties. Finally one could use the techniques called budgeting and row summing.

### 17.1.3 Unsatisfactory Optimal Solutions

Unfortunately, optimal solutions can be unrealistic. Discovering an optimal solution means the problem has a mathematically consistent optimum. However, mathematical consistency does not necessarily imply real world consistency (Heady and Candler). Usually, unrealistic solutions may be caused by improper problem specification or assumption violations. Cases arise where the model solution is improper because of: a) omitted constraints or variables; b) errors in coefficient estimation; c) algebraic errors; or d) coefficient placement errors.

Basically, a model may be judged improper because of incorrect valuation or allocation results. Valuation difficulties arise from the reduced cost or shadow price information, such items take on values when primal reduced costs are formed. Allocation difficulties arise when the slack or decision variable values are unrealistic. The values of these items are formed through the constraint interactions. Thus, to diagnose the cause of the unrealistic solution, one investigates either the reduced costs associated with the nonbasic primal variables or the calculations inherent in the primal constraints. Two techniques are presented below, one for the investigation of reduced costs, which we call "budgeting"; and another for the reconstruction of the constraint calculations, which we call "row summing."

## 17.2 Techniques for Diagnosing Improper Models

Now suppose we turn our attention to the techniques one might use to alleviate model solution difficulties. Table 17.1 presents an array of the possible problems and an indication of the techniques one might use to diagnose such problems.

### 17.2.1 Simple Structural Checking

There are some simple yet powerful techniques for checking LP formulations, regardless of their presentation method. These fall into two categories: one numerical and one analytical.

#### 17.2.1.1 Analytical Checking

In the case of analytical techniques, consider the problem:

$$\begin{aligned}
 & \text{Max} && \sum_j c_j X_j \\
 \text{s.t.} &&& \sum_j a_{ij} X_j \# b_i \quad \text{for all } i \\
 &&& \sum_j e_{nj} X_j \# d_n \quad \text{for all } n \\
 &&& \sum_j f_{mj} X_j \$ g_m \quad \text{for all } m \\
 &&& X_j \$ 0 \quad \text{for all } j
 \end{aligned}$$

Certain values of these parameters can cause the model to: 1) be infeasible, 2) contain a set of variables that must be zero, 3) contain redundant constraints, and 4) yield an unbounded solution, or 5) contain variables that are always unattractive. Table 17.2 presents a set of cases where model structures will guarantee these properties. Suppose we elaborate on one case which leads to each of the five properties.

A model formulation can cause infeasibility. Suppose in the first constraint,  $b_i$ , is less than zero and all the  $a_{ij}$ 's in that particular constraint are nonnegative. Obviously this constraint causes the model to be infeasible, since it is impossible for the sum of nonnegative numbers to be less than or equal to a negative number.

Second, it is possible that the constraints require that certain variables be zero. Consider what happens if in the second constraint the right hand side ( $d_n$ ) equals to zero and all  $e_{nj}$ 's are greater than or equal to zero, then every variable with a nonzero coefficient in that constraint must be zero.

There are also cases where the model possesses redundant constraints. Suppose  $b_i$  is positive, but all  $a_{ij}$ 's are negative or zero; then, clearly, this constraint will be redundant as the sum of negative numbers will always be less than or equal to a positive number.

Checks can also be made for whether the problem is unbounded or contains variables which will never come into the solution. Consider an activity with a positive objective function coefficient which has all nonzero  $a_{ij}$ 's negative, all zero  $e_{nj}$ 's and all nonzero  $f_{mj}$ 's positive. Clearly, then, this variable contributes revenue but relaxes all constraints. This will be unbounded regardless of the numerical values. Further, variables may be specified which will never come into the solution. For example, this is true when  $c_j$  is less than 0, all nonzero  $a_{ij}$ 's are greater than 0,  $e_{nj}$ 's zero, and nonzero  $f_{mj}$ 's negative.

These particular structural checks allow one to examine the algebraic formulation or its numerical counterpart. Unfortunately, it is not possible to make simple statements when the constraint coefficients are of mixed sign. In such cases, one will have to resort to numerical checking. All of the procedures above have been automated in GAMSCHCK although they can be programmed in GAMS (See McCarl, 1977).

#### 17.2.1.2 Numerical Model Analysis

Another model analysis methodology involves numerical investigation of the equations and variables. Here, one prints out the equations of a model (in GAMS by using the OPTION LIMROW and LIMCOL command) and mentally fixes variables at certain levels, and then examines the relationship of these variables with other variables by examining the equations. Examples of this are given in the joint products problem above. Numerical model analysis can also be carried out by making sure that units are proper, using the homogeneity of units tests.

Another numerical technique involves use of a "PICTURE" with which coefficient placement and signs can be checked. GAMS does not contain PICTURE facilities, so we do not discuss the topic here, although one is contained in GAMSCHK (see McCarl, 1977).

### **17.2.2 A Priori Degeneracy Resolution**

Degeneracy can cause solvers to cycle endlessly making little or no progress. Solvers like MINOS (Murtaugh and Saunders, 1983) on occasion give messages like "terminating since no progress made in last 1000 iterations" or "Sorry fellows we seem to be stuck." Our experience with such cases indicates one should use an a priori degeneracy resolution scheme adding small numbers to the right hand sides, especially to those constraints which start out with zero or identical right hand sides. The magnitude of the small numbers should be specified so that they are not the same for all rows and so that they do not materially affect the solution. Thus, they might be random or systematically chosen numbers of the order  $10^{-3}$  or  $10^{-4}$  (although they can be larger or smaller depending on the scaling and purpose of the constraints as in McCarl, 1977). We have always observed reduced solution times with this modification. OSL automatically invokes such a procedure.

### **17.2.3 Altering Units of Constraints and Variables: Scaling**

Scaling is done automatically in a number of algorithms including MINOS which is used in GAMS. However, automatic scaling is not always successful. Modelers are virtually always more effective in scaling (Orchard-Hayes). This section explores scaling procedures, discussing the effects on resulting optimal solutions.

Altering the units of constraints and variables improves the numerical accuracy of computer algorithms and can reduce solution time. Scaling is needed when the disparity of matrix coefficient magnitudes is large. An appropriate rule of thumb is, one should scale when the matrix coefficient magnitudes differ in magnitude by more than  $10^3$  or  $10^4$ . In other words, scaling is needed if the  $a_{ij}$  coefficient with the largest absolute value divided by the coefficient with the smallest nonzero absolute value exceeds 10000. One achieves scaling by altering the formulation so as to convert: a) the objective function to aggregate units (i.e., thousands of dollars rather than dollars), b) constraints to thousands of units rather than units (i.e., one might alter a row from pounds to tons), or c) variables into thousands of units (e.g., transport of tons rather than pounds).

#### 17.2.3.1 Scaling-The Basic Procedure

Given the LP problem

$$\begin{array}{ll}
\text{Max} & c_1 X_1 \quad \% \quad c_2 X_2 \\
\text{s.t.} & a_{11} X_1 \quad \% \quad a_{12} X_2 \quad \# \quad b_1 \\
& a_{21} X_1 \quad \% \quad a_{22} X_2 \quad \# \quad b_2 \\
& X_1, \quad X_2 \quad \$ \quad 0
\end{array}$$

Suppose one wished to change the units of a variable (for example, from pounds to thousand pounds). The homogeneity of units test requires like denominators in a column. This implies every coefficient under that variable needs to be multiplied by a scaling factor which equals the number of old variable units in the new unit; i.e., if  $X_j$  is in old units and  $X'_j$  is to be in a new unit, with  $a_{ij}$  and  $a'_{ij}$  being the associated units.

$$X'_j = X_j / SC_j;$$

where  $SC_j$  equals the scaling coefficient giving the new units over the old units

$$a'_{ij} = a_{ij} / (SC_j).$$

The scaling procedure can be demonstrated by multiplying and dividing each entry associated with the variable by the scaling factor. Suppose we scale  $X_1$  using  $SC_1$

$$\begin{array}{ll}
\text{Max} & SC_1 c_1 X_1 / SC_1 \quad \% \quad c_2 X_2 \\
\text{s.t.} & SC_1 a_{11} X_1 / SC_1 \quad \% \quad a_{12} X_2 \quad \# \quad b_1 \\
& SC_1 a_{21} X_1 / SC_1 \quad \% \quad a_{22} X_2 \quad \# \quad b_2 \\
& X_1, \quad X_2 \quad \$ \quad 0
\end{array}$$

or substituting a new variable  $X'_1 = X_1/SC_1$  we get

$$\begin{array}{ll}
\text{Max} & SC_1 c_1 X'_1 \quad \% \quad c_2 X_2 \\
\text{s.t.} & SC_1 a_{11} X'_1 \quad \% \quad a_{12} X_2 \quad \# \quad b_1 \\
& SC_1 a_{21} X'_1 \quad \% \quad a_{22} X_2 \quad \# \quad b_2 \\
& X'_1, \quad X_2 \quad \$ \quad 0
\end{array}$$

Variable scaling alters the magnitude of the solution values for the variables and their reduced cost as we will prove later.

Scaling can also be done on the constraints. When scaling constraints; e.g., transforming their units from hours to thousands of hours, every constraint coefficient is divided by the scaling factor (SR) as follows:

$$\begin{array}{ll}
\text{Max} & c_1 X_1 \quad \% \quad c_2 X_2 \\
\text{s.t.} & a_{11} / \text{SR} X_1 \quad \% \quad a_{12} / \text{SR} X_2 \quad \# \quad b_1 / \text{SR} \\
& a_{21} X_1 \quad \% \quad a_{22} X_2 \quad \# \quad b_2 \\
& X_1, \quad X_2 \quad \$ \quad 0
\end{array}$$

where SR is the number of old units in a new unit and must be positive. Constraint scaling affects :1) the slack variable solution value, which is divided by the scaling factor; 2) the reduced cost for that slack, which is multiplied by the scaling factor; and 3) the shadow price, which is multiplied by the scaling factor.

The way scaling factors are utilized may be motivated by reference to the homogeneity of units section. The coefficients associated with any variable are homogeneous in terms of their denominator units. Thus, when a variable is scaled, one multiplies all coefficients by a scaling factor (the old unit over the new unit) changing the denominator of the associated coefficients. Constraints, however, possess homogeneity of numerator units so, in scaling, we divide through by the new unit divided by the old unit. Thus, when changing a constraint from pounds to tons one divides through by 2000 (lbs/tons).

Two other types of scaling are also relevant in LP problems. Suppose that the right hand sides are scaled, i.e., scaled from single units of resources available to thousands of units of resources available. Then one would modify the model as follows:

$$\begin{array}{ll}
\text{Max} & c_1 X_1 \quad \% \quad c_2 X_2 \\
\text{s.t.} & a_{11} X_1 \quad \% \quad a_{12} X_2 \quad \# \quad b_1 / \text{SH} \\
& a_{21} X_1 \quad \% \quad a_{22} X_2 \quad \# \quad b_2 / \text{SH} \\
& X_1, \quad X_2 \quad \$ \quad 0
\end{array}$$

The net effects of this alteration will be that the optimal value of every decision variable and slack would be divided by the scaling factor, as would the optimal objective function value. The shadow prices and reduced costs would be unchanged.

One may also scale the objective function coefficients by dividing every objective function coefficient through by a uniform constant (SO).

$$\begin{array}{ll}
\text{Max} & c_1 / \text{SO} X_1 \quad \% \quad c_2 / \text{SO} X_2 \\
\text{s.t.} & a_{11} X_1 \quad \% \quad a_{12} X_2 \quad \# \quad b_1 \\
& a_{21} X_1 \quad \% \quad a_{22} X_2 \quad \# \quad b_2 \\
& X_1, \quad X_2 \quad \$ \quad 0
\end{array}$$

Under these circumstances, the optimal decision variables and slack solutions will be unchanged;



but both the shadow prices and reduced costs will be divided by the objective function scaling factor as will be the optimal objective function value.

Scaling may be done in GAMS using an undocumented feature. Namely putting in the statement `variablename.scale = 1000` would cause all variables in the named variable block to be scaled by 1000 with the solution automatically being readjusted. Similarly `equationname.scale = 1000` will scale all constraints in a block. This must be coupled with the command `Modelname.scaleopt=1`.

### 17.2.3.2 Mathematical Investigation of Scaling

In this section an investigation will be carried out on the effects of scaling using the matrix algebra optimality conditions for a linear program. Readers not interested in such rigor may wish to skip to the summary and empirical example.

The optimality conditions for the LP problem are given by

$$\begin{aligned} C_B B^{-1} a_j - c_j &\leq 0 \text{ for all } j \\ B^{-1} b &\leq 0. \end{aligned}$$

Given such a solution the optimal decision variables are given by

$$X_B = B^{-1} b,$$

the shadow prices by

$$U = C_B B^{-1}$$

and the reduced costs by

$$C_B B^{-1} a_j - c_j,$$

and the optimal Z value is

$$Z = C_B B^{-1} b$$

In our investigation, we examine the impact of scaling on each of these items.

#### 17.2.3.2.1 Variable Scaling

When a variable is scaled, the problem becomes:

$$\begin{aligned} \text{Max} \quad & SC \ c_1 \ X_1' \quad \% \ c_2 \ X_2 \\ \text{s.t.} \quad & SC \ a_{11} \ X_1' \quad \% \ a_{12} \ X_2 \ \# \ b_1 \\ & SC \ a_{21} \ X_1' \quad \% \ a_{22} \ X_2 \ \# \ b_2 \\ & X_1', \quad X_2 \ \& \ 0 \end{aligned}$$

where  $X_1'$  equals  $SCX_1$  and  $SC$  is a positive scalar.

The effect on the solution depends on whether the scaled variable is basic or nonbasic. First, consider nonbasic variables. If a nonbasic variable is scaled, then the scaling operation does not affect the basis inverse. Thus, the only thing that needs to be investigated is whether or not scaling the nonbasic

variable renders it attractive to bring into the basis. This involves an investigation of the reduced cost after scaling. Constructing the reduced cost for this particular variable

$$C_B B^{-1} SC a_j - SC c_j = SC (C_B B^{-1} a_j - c_j)$$

we find that the reduced cost after scaling (new) equals the reduced cost before scaling (old) times the scaling factor. Thus, we have the old reduced cost multiplied by the scaling constant and under positive SC the before scaling solution remains optimal. The only alteration introduced by scaling a nonbasic variable is that its reduced cost is multiplied by the scaling factor. This can be motivated practically. If it costs \$50 to enter one acre of a crop not being grown into solution, it would logically cost \$50,000 to bring in a thousand acres of that crop.

Now suppose a basic variable is scaled. In this case, the basis inverse is altered. Suppose that the basis matrix before scaling is B, while the matrix of technical coefficients before scaling is A. The new matrices (B\*,A\*) can be expressed as the old matrices (B,A) post-multiplied by matrices K<sub>A</sub> and K<sub>B</sub> which are modified identity matrices. Assuming the n<sup>th</sup> column of B is being scaled, then the element on the diagonal in the n<sup>th</sup> column of the K<sub>B</sub> matrix will be the scaling factor. Thus,

$$A^* = AK_A \quad B^* = BK_B$$

where

$$K_B^{-1} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & SC & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & \dots & 1 \end{bmatrix}$$

The K<sub>A</sub> matrix would be formed similarly with the column in the A matrix being scaled identifying the diagonal element where SC appears.

We may derive a relationship between the basis inverses before and after scaling. Matrix algebra theory shows that

$$(B^*)^{-1} = (BK_B)^{-1} = K_B^{-1} B^{-1}$$

We should also note that the scaled objective function coefficients of the basic variables are post-multiplied by K<sub>B</sub>, i.e.,

$$C_B = C_B K_B$$

Now let us look at the optimality criteria for non-basic variables

$$C_B' B^{-1} a_j' & c_j'$$

The reduced cost after scaling becomes

$$C_B' B^{-1} a_j' & c_j' = C_B K_B K_B^{-1} B^{-1} a_j' & c_j'$$

$$= C_B B^{-1} a_j' & c_j' \geq 0$$

since  $K_B K_B^{-1} = I$ . Thus, the reduced costs after scaling equal the reduced costs before scaling. Thus, we have proven that the solution will remain optimal.

We now need to turn our attention to whether or not the basic variables remain nonnegative. The values of the basic variables at optimality are given by  $B^{-1}b$ . Substituting in our relationships, we obtain

$$X_B' = B^{-1} b' = K_B^{-1} B^{-1} b' = K_B^{-1} X_B'$$

The scaled solution equals  $K_B^{-1}$  times the unscaled solution. The inverse of  $K_B$  is an identity-like matrix with one over the scaling factor on the diagonal in the position where the scaled variable enters the basis.

$$K_B^{-1} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1/SC & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & \dots & 1 \end{bmatrix}$$

The  $K_B^{-1} X_B$  multiplication yields the vector

$$X_B' = K_B^{-1} X_B = \begin{bmatrix} X_{B_1} \\ X_{B_2} \\ \dots \\ X_{B_k} / SC \\ \dots \\ X_{B_m} \end{bmatrix}$$

Thus, scaling reduces the magnitude of the particular basic variable being scaled, while all other variables are unaffected.

We may also investigate the objective function consequences. The optimal objective function value

of the scaled problem is

$$Z' = C_B' B'^{-1} b' - C_B' K_B K_B^{-1} B'^{-1} b' - C_B' B'^{-1} b' - Z$$

Clearly, then, the objective function value after the change equals the objective function value before the change. All in all, column scaling leaves the problem with the same qualitative answer. The solution value of the particular variable being scaled and its reduced cost are altered by the scaling factor.

### 17.2.3.2.2 Effects of Constraint Scaling

When one scales a constraint, the resultant problem appears as

$$\begin{aligned} \text{Max} \quad & c_1 X_1 \quad \text{\%} \quad c_2 X_2 \\ \text{s.t.} \quad & a_{11} / \text{SR} X_1 \quad \text{\%} \quad a_{12} / \text{SR} X_2 \quad \# \quad b_1 / \text{SR} \\ & a_{21} X_1 \quad \text{\%} \quad a_{22} X_2 \quad \# \quad b_2 \\ & X_1 \quad , \quad X_2 \quad \$ \quad 0 \end{aligned}$$

or

$$\begin{aligned} \text{Max} \quad & C X \\ \text{s.t.} \quad & RAX \quad \# \quad Rb \\ & X \quad \$ \quad 0 \end{aligned} \quad \text{or} \quad \begin{aligned} \text{Max} \quad & C X \\ \text{s.t.} \quad & A' X \quad \# \quad b' \\ & X \quad \$ \quad 0, \end{aligned}$$

where R is a scaling matrix of the form

$$R^{-1} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1/\text{RS} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & 1 \end{bmatrix}$$

Further, the new basis ( $B^*$ ) is related to the old basis as follows

$$B^* = RB$$

and the basis inverse is the old basis inverse multiplied by the inverse of R.

$$B^{*-1} = B^{-1} R^{-1}$$

Again,  $R^{-1}$  is an identity-like matrix quality.

$$R^{-1} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & RS & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & \dots & 1 \end{pmatrix}$$

Now let us turn our attention to the effects of scaling a constraint. We will derive the results assuming the slack variable is not in the basis. The reduced cost criteria for the scaled problem is given by

$$C_B - B^{-1} a_j \text{ \& } c_j - c_j = C_B - B^{-1} R^{-1} R a_j \text{ \& } c_j - c_j = C_B - B^{-1} a_j \text{ \& } c_j.$$

Thus, the optimality conditions are the same as before scaling and the solution remains optimal with unchanged reduced costs. We now may investigate the variable solution values. For the scaled problem, the solution values are

$$X_B^* = B^{*-1} b^* = B^{-1} R^{-1} R b = B^{-1} b$$

which shows that the values of the basic variables are unaffected. The objective function is also unchanged.

Thus, the optimality and feasibility of the basis remain entirely unaffected. What then does change? The shadow prices after scaling are

$$U = C_B B^{-1} R^{-1}.$$

Given the form of  $R^{-1}$  from above, the shadow prices are identical to the shadow prices before scaling

$$U_i = (C_B B^{-1})_i$$

for all rows but the particular row being scaled. For that row, the shadow price is multiplied by the scaling factor

$$U_i = (C_B B^{-1})_i RS$$

Finally we should note that when a slack variable is in the basis, then constraint scaling simply changes the magnitude of the optimal slack variable value by dividing it by the scaling factor.

### 17.2.3.2.3 Objective Function and Right Hand Side Scaling

If the objective function coefficients are uniformly divided by a constant, the values of the solution variables ( $B^{-1} b$ ) are unaffected. However, the magnitudes of the shadow prices ( $C_B B^{-1}$ ), the optimal objective function value ( $C_B B^{-1} b$ ), and the reduced costs ( $C_B B^{-1} a_j - c_j$ ) are affected. In all of these cases, these items would be uniformly divided by the objective function scaling factor. A similar observation can be made regarding scaling the right hand side. The right hand side determines only the objective function

value  $(C_B B^{-1} b)$  and the solution value of the basic variables  $(B^{-1} b)$ . Dividing all right hand sides by a constant would divide the objective function and all the optimal variable values by the same constant.

#### 17.2.3.4 Summary

Scaling alters coefficient magnitudes within the matrix and the resultant magnitude of selected

$$\begin{array}{ll} \text{Max} & CX \\ \text{s.t.} & AX \leq b \\ & X \geq 0 \end{array}$$

items in the solution. Consider the LP problem

Suppose a set of positive (all scaling factors must be positive) scaling factors will be applied to the LP model. The scaling factors are a) COLSCAL<sub>j</sub> for the j<sup>th</sup> variable - a factor multiplying every coefficient under that variable, b) ROWSCAL<sub>i</sub> for the i<sup>th</sup> constraint - a factor dividing every coefficient in that constraint, c) OBJSCAL for the objective function - a factor dividing every coefficient in the objective row and d) RHSSCAL for the right hand side - a factor dividing every right hand side value. The parameters of the model after scaling are:

$$\begin{array}{l} c'_j = c_j \left( \frac{\text{COLSCAL}_j}{\text{OBJSCAL}} \right) \\ a'_{ij} = a_{ij} \left( \frac{\text{COLSCAL}_j}{\text{ROWSCAL}_i} \right) \\ b'_i = b_i \left( \frac{1}{\text{RHSSCAL}_i} \right) \end{array}$$

where the ' denotes the new coefficients. The relationship between solution items before and after scaling is given in Table 17.3. Thus, if a particular variable j is scaled by 1000 and constraint i is scaled by a 1000, the a<sub>ij</sub> value is numerically unchanged. However, if variable j is scaled by 1000 and constraint i was scaled by 100, then the value of the a<sub>ij</sub> coefficient is multiplied by 10.

Thus, for example, if an optimal variable value before scaling was 6 and the right hand side is multiplied by 100 while the coefficients of that variable are multiplied by .02 then the resultant value after scaling would be 3.

Finally, we must emphasize that the only proper way of scaling is to operate on all coefficients for each variable, right hand side and objective function in the same manner. One cannot selectively scale selected coefficients. The interaction of the various scaling factors can make it look like one is only scaling selected coefficients, as will be demonstrated in the empirical example below. But this is not the case, consistency must be maintained.

#### 17.2.3.5 Empirical Example of Scaling

The previous discussion deals with the implications of changes in constraint, variable, right hand

side and objective function units. The reader, however, should note that gains in terms of numerical stability may arise only when several items are simultaneously scaled. This is illustrated in the example.

Consider the following problem

$$\begin{array}{rllllll}
 \text{Max} & X_1 & & 500X_2 & & 400X_3 & & 5000X_4 & & \\
 \text{s.t.} & X_1 & & 10000X_2 & & 8000X_3 & & & \# & 0 \\
 & & & 5X_2 & \% & 4X_3 & & 50X_4 & \# & 0 \\
 & & & 1500X_2 & \% & 2000X_3 & & & \# & 600000 \\
 & & & 50X_2 & \% & 45X_3 & & & \# & 30000 \\
 & X_1, & & X_2, & & X_3, & & X_4 & \$ & 0
 \end{array}$$

where:

- $X_1$  is the sale of nibbles in pounds. It returns \$1 of profit per unit and removes one pound from the nibble balance row.
- $X_2$  is the hours of nibble production via process 1. One hour's worth of production uses \$500 worth of direct cost, 5 units of gribbles, 1500 hibbles and 50 hours of labor. As a result, one gets 10,000 nibbles.
- $X_3$  is the hours of nibble production using process 2. Here 8000 nibbles are produced at a direct cost of \$400 with four pounds of gribbles used, 2,000 hibbles, and 45 hours of labor produced.
- $X_4$  is the number of 50 pound sacks of gribbles purchased, costing \$5000 and providing 50 pounds of gribbles into the gribble balance row.

The right hand side shows an endowment of 600,000 hibbles and 30,000 hours of labor. The objective function is in the units of dollars and represents profit. The first constraint balances the units of nibbles produced with those sold. The second constraint balances the units of gribbles used with those purchased. The third constraint limits the number of hibbles used to the fixed endowment. The fourth constraint limits the hours of labor used to the fixed endowment. Non-negativity of all variables is assumed.

This problem is not well-scaled and its scaling characteristics will be altered. (This will be done to illustrate scaling - the problem is scaled satisfactorily for any solver). At solution, the objective function equals 3,600,000 and the variables values (with their units) are shown in Table 17.4.

Now suppose we scale the first constraint by dividing through by 1000. Simultaneously, let us scale the third constraint by dividing through by 100, and divide the fourth constraint by 10. This changes the units of these constraints such that the first constraint is in thousands of nibbles, the third constraint is







labor endowment is erroneously too small, c) the labor requirements for production have been overestimated, or d) the contribution to the minimum requirement constraint is too small.

The last three cases arise when the minimum requirement level is correct, but the infeasibility is caused by misspecifications in other coefficients. Thus, infeasibilities arise not only because of improperly specified minimum requirement rows, but also because of errors in other coefficients.

The question is "how can one discover the cause of the infeasibility?" This can be done by adding artificial variables to the model formulation. The inclusion of artificial variables permits all models to have feasible optimal solutions regardless of whether the "real" constraints are satisfied. Infeasible solutions exhibit nonzero artificial variables. Nonzero artificial variables will cause a large negative objective function value and large shadow prices since some  $C_B$ 's in the  $C_B B^{-1}$  computations are large. Specifically, constraints which exhibit large shadow prices are those involved with the infeasibility. The constraints not causing the infeasibility will have unaffected shadow prices. Thus, the imposition of artificial variables allows one to identify which constraints are nominally causing the infeasibility. We do not argue that such information cannot be found in an ordinary infeasible solution; however, it is more difficult to interpret. Ordinarily, infeasible solver solutions are detected by phase 1 of the simplex algorithm wherein the shadow prices give the marginal contribution of a change in the right hand side to the sum of the infeasibilities.

To illustrate the use of artificial variables in the context of an infeasible model consider the following example:

$$\begin{array}{rcll}
 \text{Max} & 50X_1 & \% & 50X_2 \\
 \text{s.t.} & X_1 & \% & X_2 \# 50 \\
 & 50X_1 & \% & X_2 \# 65 \\
 & X_1 & & \$ 20 \\
 & X_1 & , & X_2 \$ 0
 \end{array}$$

This problem is infeasible due to the interaction of the second and third constraints. Suppose that an error was made and the number 50 which is specified as the requirement of  $X_1$  for the second resource should have been 0.50. The third constraint has a minimum requirement, thus an artificial variable is included.

$$\begin{array}{rcll}
 \text{Max} & 50X_1 & \% & 50X_2 & \& & 10000A \\
 & X_1 & \% & X_2 & & \# & 50 \\
 & 50X_1 & \% & X_2 & & \# & 65 \\
 & X_1 & & & \% & A & \$ 20
 \end{array}$$

Here the artificial variable A is entered with a large negative number in the objective function and a



\$2 without limit. However, the imposition of the constraint  $X_1 \leq 100,000$  yields the solution  $X_1 = X_2 = 100,000$ ,  $X_3 = 50$ . Thus, if the model user saw this solution and felt that  $X_1 = 100,000$  was unrealistically large then this would show that there is something wrong within the model. It also shows that  $X_1$  and  $X_2$  are the items involved with the unboundedness while  $X_3$  is not a factor.

The use of large upper bounds precludes the possibility of an unbounded solution but causes the objective function and some of the variables to take on unrealistically large values. Subsequently, one can trace the cause of the unboundedness by examining the set of variables which are unrealistically large. This is important since ordinarily LP solvers are implemented so that when they discover an unbounded solution they automatically stop. This leaves the user without much information as to the cause of the unboundedness.

Decision modelers may wish to upper bound the sum of a number of variables rather than each and every variable. This could be done by using the following:

$$\begin{aligned} \text{Max } & \sum_j c_j X_j \\ \text{s.t. } & \sum_j a_{ij} X_j \leq b_i \text{ for all } i \\ & \sum_j X_j \leq M \\ & X_j \geq 0 \text{ for all } j \end{aligned}$$

Here one specifies that the sum of all variables is less than or equal to an unrealistically large number.

### 17.2.6 Budgeting

Yet another model analysis technique, particularly when dealing with unrealistic optimal solutions, involves Budgeting. Budgeting herein refers to the reconstruction and examination of reduced cost and shadow price information. The procedure is best illustrated through example. Consider the model shown in Table 17.9.

This model contains activities for buying miscellaneous inputs; selling corn, soybeans, and pork; and producing corn, soybeans, and hogs. The model is maximized subject to a resource constraint on land, along with supply-demand balances on pork, soybeans, corn, and miscellaneous inputs. The miscellaneous input item is specified in dollars and therefore enters the objective function at a per unit cost of \$1 while supplying a dollar's worth of miscellaneous inputs. Corn is sold for \$2.50 per unit, soybeans \$6 per unit, and pork \$.50 per unit. Corn production incurs \$75 in direct production costs and \$125 in miscellaneous inputs while using one acre of land and yielding 120 bushels of corn. Soybean production costs \$50 in direct production costs and another \$50 in miscellaneous inputs while using an acre of land and yielding 50

bushels of soybeans. Hog production has no direct costs, uses \$20 in miscellaneous inputs, and requires 20 bushels of corn. An unrealistically large yield in the hog activity has been entered (1000 pounds per hog). This example "error" will be sought by the budgeting technique.

The optimum solution to this model is shown in Table 17.10. The optimal value of the objective function is \$1,508,000. This solution includes several symptoms that there is something wrong. For example, 3,600,000 pounds of pork are sold, the reduced cost on raising soybeans is \$2,480 an acre, the shadow price on land is \$2,680 and the shadow price of corn is \$24 a bushel. Budgeting investigates the shadow prices and reduced costs in an effort to discover model misspecifications.

The matrix and summation formula for reduced costs is

$$C_B B^{-1} a_j - c_j - \sum_i (C_B B^{-1})_i a_{ij} + c_j \leq 0,$$

which will be greater than or equal to zero for all nonbasic variables and exactly equal to zero for basic variables. Utilizing the fact that the  $C_B B^{-1}$  are the shadow prices, then the equation becomes

$$\sum_i U_i a_{ij} - c_j \leq 0$$

where  $U_i$  is the shadow price associated with row  $i$ , the  $a_{ij}$ 's are the technical coefficients of the original model, and the  $c_j$  is the original objective function coefficients associated with  $X_j$ .

Budgeting involves construction of an extensive version of the reduced cost calculations which in turn are examined for plausibility. The variable budgeted first could be chosen because: a) it is nonbasic when intuition suggests it should be basic; b) it has an unrealistically high reduced cost; or c) it uses a resource which appears to be improperly valued. In the example, suppose we budget soybean production because of its high reduced cost. To budget, write a row for each nonzero coefficient ( $a_{ij}$ ) under the chosen variable, with associated shadow prices ( $U_i$ ) and  $a_{ij}U_i$  the product, then finally subtracting cost. The budget for soybean production is shown in Table 17.11.

Mechanically the budget examines the cost of resource usage in those rows for which the activity uses resources and values (shadow price) of these resources. In turn the objective function value of the variable is considered and the reduced costs reconstructed. The soybean production variable has non-zero coefficients in the land, soybean production, and miscellaneous input constraints. The shadow price for land is \$2,680. Thus, one acre of soybeans uses \$2,680 worth of land and yields 50 bushels, each selling for \$6. Also, 50 units of miscellaneous inputs are used which, when valued at \$1, cost \$50. Summing these terms, the marginal contribution of soybean production, ignoring its direct costs, is \$2,430. Its direct cost ( $c_j=50$ ) is then subtracted yielding a \$2,480 reduced cost. One may conclude that, the \$2,480 reduced cost is caused by the \$2,680 shadow price on land. The question then becomes why is land this valuable.

Shadow prices are derived from setting the reduced costs of basic variables to zero. Thus, the high land shadow price must arise from the reduced costs of some basic variable which utilizes land. The only land using basic variable is corn production. We then budget the corn production variable (Table 17.12). Note that while one acre of corn production uses \$2,680 of land, it receives \$2,880 from the value of the corn sold. Here, the reason for the \$2,680 cost of land is the \$2,880 value of the corn. Institutional knowledge indicates the 120 bushels per acre corn yield is reasonable, but the \$24 corn shadow price per bushel is not. Thus, the question becomes, "Why is the corn shadow price so high?" Again, this will be determined by a basic variable which utilizes corn. The only basic cornusing variable is hog production. The budget for hog production is shown in Table 17.13. These computations show that zero reduced cost for this activity requires that 20 bushels of corn be valued at \$24/unit. The cause of the \$500/bushel value for corn is an unrealistic value of pork produced (\$500). The erroneous 1000 pound coefficient for pork production per hog would then be discovered. A revised value of the pork yield per hog would alter the model, making the solution more realistic.

The budgeting technique is useful in a number of settings. Through its use, one may discover why variables are nonbasic when they should be basic. The soybean production variable budget provides such an example. Budgeting, in such a case, may discover difficulties in the particular variable being budgeted or in shadow prices.

Budgeting may also be used to discover why particular activities are basic when modeler intuition suggests they should be nonbasic. For example, by tracing out the costs and returns to corn as opposed to soybean production to see what the major differences that lead to corn being profitable while soybeans are not.

The third use of budgeting involves discovering the causes of improper shadow prices. Shadow prices arise from a residual accounting framework where, after the fixed revenues and costs are considered, the residual income is attributed to the unpriced resources.

Budgeting can also be used to deal with infeasible solutions from Phase I of a Phase I/Phase II simplex algorithm. Phase I of such algorithms minimizes the sum of infeasibilities. Thus, all of the objective function coefficients of the decision variables in the model are set to zero. The phase I shadow prices refer to the amount by which the sum of the infeasibilities will be reduced by a change in the right hand sides. Budgeting then can be done to trace shadow price origins and to see why certain variables do not come into solution. Solutions containing artificial variables may also be budgeted.

### **17.2.7 Row Summing**

Model solutions also may be analyzed by examination of the primal allocation results. In the budgeting example problem, one could have examined the reasons for the sale of 3.6 million pounds of

pork. This can be done through a procedure we call row summing. This is illustrated through a slightly different, but related, example Table 17.14.

Compared to the model shown in Table 17.9, the pork production coefficient has been altered to -150, while the corn yield per unit has been changed to an incorrect value of -1200 -- the error. We have also introduced a RHS of 20 on the corn balance equation. The solution to this model is shown in Table 17.15. The optimal value of the objective function is \$1,860,055. Here 5.4 million pounds of pork are sold which one would probably judge to be unrealistically high. Further, there are more than 36,000 hogs on the farm.

A row sum is simply a detailed breakdown of a constraint: each variable appearing in that constraint, its corresponding coefficient ( $a_{ij}$ ) and the product  $a_{ij}X_j$ . The products are then summed, and subtracted from the right hand side and the slack variable formed. The use of row summing in our example begins with the pork sales constraint to see if 5.4 million lbs. is reasonable (Table 17.16.).

The pork constraint contains the variables sell pork and hog production. The sell pork variable uses one pound of pork per unit, while the hog production variable yields 150 pounds of pork per unit. The second column of Table 17.15 contains the optimal variable values. In the third column we write the product of the variable value and its  $a_{ij}$ . The products are summed to give total endogenous use which in this case equals zero. We then enter the right hand side and subtract it to determine the value of the slack variable. All these items in this case are zero. Given institutional knowledge, one would conclude the error has not yet been found as the 150 lbs. of pork per hog is reasonable, and all pork produced is sold. However, one would wonder if a production level of 36,001 hogs is reasonable. The next step is to examine the resources used by hog production. For illustrative purposes, we begin with the miscellaneous input supply-demand balance. The row sum for this constraint is shown in Table 17.17.

There are four entries in the constraint involving both basic and nonbasic variables. The row sum does not reveal anything terribly unrealistic except the large amount of activity from the hog production variable. The basic question is yet to be resolved.

We next investigate the corn supply-demand balance. The row sum computations for this constraint are shown in Table 17.18. In this case the constraint has a non-zero right hand side; thus, the endogenous sum is 20 which equals the right hand side leaving the slack variable zero. We find the 36,001 hogs require 720,020 bushels of corn, and the reason they are able to obtain all this corn is because of the inaccurate yield on the corn production variable. The modeler would then correct the yield on the corn production variable.

The above example illustrates the principles behind using the allocation results to debug a model.

One identifies a variable or slack with an unrealistically high solution value, and then row sums the constraints in which that variable is involved with to discover the problem. Row summing can be used to discover why particular variables have unrealistically large values by identifying incorrect coefficient values or coefficient placement errors. For example, suppose that the corn yield was inadvertently punched in the soybean row; then one might have discovered a solution in which soybeans are sold but no soybeans are produced. A row sum would quickly determine the source of the soybeans and indicate the error. Row summing can also be applied to discover the causes of large values for slack or surplus variables.



## References

- Heady, E.O. and W.V. Candler. Linear Programming Methods. Iowa State University Press: Ames, Iowa, 1958.
- McCarl, B.A. "Degeneracy, Duality, and Shadow Prices in Linear Programming." Canadian Journal of Agricultural Economics. 25(1977):70-73.
- Murtaugh, B. and M. Saunders. "MINOS 5.0 Users Guide." Technical Report SOL 83-20 Stanford University, 1983.
- Orchard, Hays W. Advanced Linear Programming Computing Techniques. McGraw-Hill: New York, 1968.

**Table 17.1. Priorities of Techniques to Use to Diagnose Improper Model Solution Outcomes**

of Solution Outcome	Type	Structural Check <sup>a</sup>	Degen. Resol.	Scaling <sup>a</sup>	Artificial Variables	Upper Bounds	Budget	Row Sum
Solver Failure		1	3	2	5	4		
Unbounded Solution		1		3		2	4	
Infeasible Solutions		1		3	2		4	5
Unsat. Optimal Solutions		1					2	2

Notes: The entries in the table gives information on the order in which to try techniques with the technique numbered 1 being the item to try first.

<sup>a</sup> This technique could be employed before any solving occurs. The technique also can be used when problems appear.

**Table 17.2. Solution Properties of Various Model Formulations**

---

**Cases Where the Model Must have an Infeasible Solution**

$b_i < 0$  and  $a_{ij} \leq 0$  for all  $j \forall$  row  $i$  will not allow a feasible solution

$d_n < 0$  and  $e_{nj} \leq 0$  for all  $j \forall$  row  $n$  will not allow a feasible solution

$d_n > 0$  and  $e_{nj} \neq 0$  for all  $j \forall$  row  $n$  will not allow a feasible solution

$g_m > 0$  and  $f_{mj} \neq 0$  for all  $j \forall$  row  $m$  will not allow a feasible solution

**Cases where certain variables in the model must equal zero**

$b_i = 0$  and  $a_{ij} \leq 0$  for all  $j \forall$  all  $X_j$ 's with  $a_{ij} \hat{=} 0$  in row  $i$  will be zero

$d_n = 0$  and  $e_{nj} \leq 0$  for all  $j \forall$  all  $X_j$ 's with  $e_{nj} \hat{=} 0$  in row  $n$  will be zero

$d_n = 0$  and  $e_{nj} \neq 0$  for all  $j \forall$  all  $X_j$ 's with  $e_{nj} \hat{=} 0$  in row  $n$  will be zero

$g_m = 0$  and  $f_{mj} \neq 0$  for all  $j \forall$  all  $X_j$ 's with  $f_{mj} \hat{=} 0$  in row  $m$  will be zero

**Cases where certain constraints are obviously redundant**

$b_i \leq 0$  and  $a_{ij} \neq 0$  for all  $j$  means row  $i$  is redundant

$g_m \neq 0$  and  $f_{mj} \leq 0$  for all  $j$  means row  $m$  is redundant

**Cases where certain variables cause the model to be unbounded**

$c_j > 0$  and  $a_{ij} \neq 0$  or  $e_{nj} = 0$  and  $f_{mj} \leq 0$  for all  $i, m,$  and  $n$  means variable  $j$  is unbounded

**Cases where certain variables will be zero at optimality**

$c_j < 0$  and  $a_{ij} \leq 0$  or  $e_{nj} = 0$  and  $f_{mj} \neq 0$  for all  $i, m,$  and  $n$  means variable  $j$  will always be zero

---

**Table 17.3. Relationships Between Items Before and After Scaling**

Item	Symbol Before Scaling	Symbol After Scaling	Unscaled Value in Terms of Scaled Value	Scaled Value in Terms of Unscaled Value
Variables	$X_j$	$X'_j$	$X_j = X'_j * (\text{COLSCAL}_j * \text{RHSSCAL})$	$X'_j = X_j / (\text{COLSCAL}_j * \text{RHSSCAL})$
Slacks	$S_i$	$S'_i$	$S_i = S'_i * (\text{ROWSCAL}_i * \text{RHSSCAL})$	$S'_i = S_i / (\text{ROWSCAL}_i * \text{RHSSCAL})$
Reduced Cost	$z_j - c_j$	$z'_j - c'_j$	$z_j - c_j = (z'_j - c'_j) * (\text{OBJSCAL}/\text{COLSCAL}_j)$	$z'_j - c'_j = (z_j - c_j) / (\text{OBJSCAL}/\text{COLSCAL}_j)$
Shadow Price	$U_i$	$U'_i$	$U_i = U'_i * (\text{OBJSCAL}/\text{ROWSCAL}_i)$	$U'_i = U_i / (\text{OBJSCAL}/\text{ROWSCAL}_i)$
Obj. Func. Value	$Z$	$Z'$	$Z = Z' * \text{OBJSCAL} * \text{RHSSCAL}$	$Z' = Z / (\text{OBJSCAL} * \text{RHSSCAL})$

**Table 17.4. Optimal Solution to Unscaled Nibble Production Problem**

Obj = 3,600,000							
Variable	Units	Value	Reduced Cost	Equation	Unit	Slack	Shadow Price
X <sub>1</sub>	Lbs. of Nibbles	4000000	0	1	Lbs. of Nibbles	0	1
X <sub>2</sub>	Hrs. of Process 1	400	0	2	Lbs. of Gribbles	0	100
X <sub>3</sub>	Hrs. of Process 2	0	4800	3	# of Hibbles	0	6
X <sub>4</sub>	Sacks of Gribbles	40	0	4	Hrs of Labor	10000	0

**Table 17.5. Optimal Solution to Nibble Production Problem After Row Scaling**

Variable	Units	Value	Reduced Cost	Equation	Unit	Slack	Shadow Price
X <sub>1</sub>	Lbs. of Nibbles	4000000	0	1	1000's of Lbs. of Nibbles	0	1000
X <sub>2</sub>	Hrs. of Process 1	400	0	2	Lbs. of Gribbles	0	100
X <sub>3</sub>	Hrs. of Process 2	0	4800	3	100's of Hibbles	0	600
X <sub>4</sub>	Sacks of Gribbles	40	0	4	10's of Hrs of Labor	10000	0

**Table 17.6. Optimal Solution to Nibble Production Problem After Row and Column Scaling**

Variable	Units	Value	Reduced Cost	Equation	Unit	Slack	Shadow Price
X <sub>1</sub>	1000's of Lbs. of Nibbles	4000000	0	1	1000's of Lbs. of Nibbles	0	1000
X <sub>2</sub>	Hrs. of Process 1	400	0	2	Lbs. of Gribbles	0	100
X <sub>3</sub>	Hrs. of Process 2	0	4800	3	100's of Hibbles	0	600
X <sub>4</sub>	Sacks of Gribbles	40	0	4	10's of Hrs of Labor	10000	0

**Table 17.7. Optimal Solution to Nibble Production Problem After Row, Column, Objective Function and RHS Scaling**

Variable	Units	Value	Reduced Cost	Equation	Unit	Slack	Shadow Price
X <sub>1</sub>	100,000's Lbs. of Nibbles	40	0	1	100,000's of Lbs. of Nibbles	0	1
X <sub>2</sub>	100's of Hrs. of Process 1	4	0	2	100's Lbs. of Gribbles	0	0.1
X <sub>3</sub>	100's of Hrs. of Process 2	0	4.8	3	10,000's of Hibbles	0	0.6
X <sub>4</sub>	100's of Sacks of Gribbles	20	0	4	1000's of Hrs of Labor	100	0

**Table 17.8. Solution to Infeasible Example with Artificial Present**

Variable	Value	Reduced Cost	Equation	Level	Shadow Price
X <sub>1</sub>	1.3	0	1	48.7	0
X <sub>2</sub>	0	1940	2	0	1990
A	18.7	0	3	0	-10,000

Objective Function = -186935

**Table 17.9. Tableau of Budgeting Example**

Row	Buy Misc.	Sell Corn	Sell Soyb.	Sell Pork	Prod Corn	Prod Soyb.	Prod Hogs	RHS
Objective Func	-1	2.5	6	0.5	-75	-50		MAX
Land Available					1	1		# 600
Pork Balance			1				-1000	# 0
Soybean Bal			1			-50		# 0
Corn Balance		1			-120		20	# 0
Misc. Inp. Bal.	-1				125	50	20	# 0

**Table 17.10. Optimal Solution to Budgeting Example**

Variable	Value	Reduced Cost	Equation	Level	Shadow Price
Buy Misc. Input	147,000	0	Land Available	0	2680.00
Sell Corn	0	22.50	Pork Balance	0	0.5
Sell Soybeans	0	0	Soybean Balance	0	6.00
Sell Pork	3,600,000	0	Corn Balance	0	24.00
Produce Corn	600	0	Misc. Input Balance	0	1.00
Produce Soybeans	0	2,480.00			
Produce Hogs	3,600	0			

**Table 17.11. Budget of Soybean Production Activity**

Constraint	$a_{ij}$	Shadow Price ( $U_j$ )	Product ( $U_i a_{ij}$ )
Land Available	1	2680	2680
Soybean Balance	-50	6	-300
Misc. Input Balance	50	1	50
Indirect Cost Sum ( $\sum U_i a_{ij}$ )			2430
Less Objective Function ( $c_j$ )	-50		-(-50)
Red. Cost ( $\sum U_i a_{ij} - c_j$ )			2480(nonbasic)



**Table 17.12. Corn Production Budget**

Constraint	$a_{ij}$	Shadow Price ( $U_j$ )	Product ( $U_j a_{ij}$ )
Land Available	1	2680	2680
Corn Balance	-120	24	-2880
Misc. Input Balance	125	1	125
Indirect Cost Sum ( $\sum U_i a_{ij}$ )			-75
Less Objective Function ( $c_j$ )	-75		-(-75)
Reduced Cost ( $\sum U_j a_{ij} - c_j$ )			0(basic)

**Table 17.13. Hog Production Budget**

Constraint	$a_{ij}$	Shadow Price ( $U_j$ )	Product ( $U_j a_{ij}$ )
Pork Balance	-1000	0.5	-500
Corn Balance	20	24	480
Misc. Input Balance	20	1	20
Indirect Cost Sum ( $\sum U_j a_{ij}$ )			0
Less Objective Function ( $c_j$ )	0		-(0)
Reduced Cost ( $\sum U_j a_{ij} - c_j$ )			0(basic)

Row	Buy Misc.	Sell Corn	Sell Soyb.	Sell Pork	Prod Corn	Prod Soyb.	Prod Hogs	RHS
Objective Func	-1	2.5	6	0.5	-75	-50		MAX
Land Available					1	1		# 600
Pork Balance			1	1			-150	# 0
Soybean Bal			1			-50		# 0
Corn Balance		1			-1200	20		# 20
Misc. Inp. Bal.	-1				125	50	20	# 0

**Table 17.15. Optimal Solution to Row Summing Example**

Variable	Value	Reduced Cost	Equation	Level	Shadow Price
Buy Misc. Input	795,020	0	Land Available	0	3,100
Sell Corn	0	0.25	Pork Balance	0	0.5
Sell Soybeans	0	0	Soybean Balance	0	6.00
Sell Pork	5,400,150	0	Corn Balance	0	2.75
Produce Corn	600	0	Misc. Input Balance	0	1.00
Produce Soybeans	0	2,480.00			
Produce Hogs	36,001	0			

**Table 17.16. Row Sum of Pork Constraint**

Variable	$a_{ij}$	Optimal Value ( $X_j^*$ )	Product ( $a_{ij}X_j^*$ )
Sell Pork	1	5,400,150	5,400,150
Produce Hogs	-150	36,001	-5,400,150
Endogenous Sum ( $\sum a_{ij} X_j^*$ )			0
Right Hand Side( $b_i$ )	0		0
Slack ( $b_i - \sum a_{ij} X_j^*$ )			0

**Table 17.17. Row Sum of Miscellaneous Input Constraint**

Variable	$a_{ij}$	Optimal Value ( $X_j^*$ )	Product ( $a_{ij}X_j^*$ )
Buy Miscellaneous Inputs	-1	795,020	-795,020
Produce Corn	125	600	75,000
Produce Soybeans	50	0	0
Produce Hogs	20	36,001	720,020
Endogenous Sum ( $\sum a_{ij} X_j^*$ )			0
Right Hand Side( $b_i$ )	0		0
Slack ( $b_i - \sum a_{ij} X_j^*$ )			0

**Table 17.18. Row Sum of Corn Balance Constraint**

Variable	$a_{ij}$	Optimal Value ( $X_j^*$ )	Product ( $a_{ij}X_j^*$ )
Sell Corn	1	0	0
Produce Corn	-1,200	600	-720,000
Produce Hogs	20	36,001	720,020
Endogenous Sum ( $\sum a_{ij} X_j^*$ )			20
Right Hand Side( $b_i$ )	20		20
Slack ( $b_i - \sum a_{ij} X_j^*$ )			0

## CHAPTER XVIII: VALIDATION OF PROGRAMMING MODELS

Model validation is important in any empirical analysis<sup>14</sup>. Programming models frequently are superficially validated. However, validation is necessary for both predictive and prescriptive model use<sup>15</sup>. Validation exercises almost always improve model performance and problem insight.

This chapter presents procedures for programming model validation and cites examples. The discussion will be most relevant to predictive model validation, however, the procedures may also be used with prescriptive models.

### 18.1 Background

Before beginning the presentation, a model structure is needed. Let the model contain demand(X), production(Y) and input purchase variables( Z) with the following structure.

$$\begin{array}{rcl}
 \text{Max} & f(X) & \& g(Z) \\
 \text{s.t.} & X & \& DY & \# & 0 \\
 & & & GY & \& Z & \# & 0 \\
 & & & AY & \# & b \\
 & X & , & Y & , & Z & \# & 0
 \end{array}$$

Let us denote the optimal values of these variables as  $X^*$ ,  $Y^*$ ,  $Z^*$ . Now suppose these variables are assumed to correspond to real world observations  $\bar{X}$ ,  $\bar{Y}$ , and  $\bar{Z}$ . The model also has associated shadow prices, U, V, and W which at optimality are  $U^*$ ,  $V^*$ , and  $W^*$  and correspond to real world observations  $\bar{U}$ ,  $\bar{V}$ , and  $\bar{W}$ .

### 18.2 General Approaches to Validation

Validation approaches vary widely. The overall purpose is to test how well a model serves its intended purpose. For predictive models, validation tests can involve comparing model predictions to real world results. For prescriptive models, decision maker reliance is the ultimate validation test. Unfortunately, these tests can rarely be used because they are expensive and time-consuming (this is often the reason for modeling in the first place). Thus, models are frequently validated using historical events. Although a model may have a broad range of potential uses, it may be valid only for a few of those uses. The validation process usually results in identification of valid applications.

---

<sup>14</sup> The material in this chapter is largely drawn from McCarl (1984) and McCarl and Apland.

<sup>15</sup> The word validate is controversial. Some prefer to use verify. Within this text, validate refers to exercises determining whether the model user and or modeling team feels the model behavior is close enough to real world behavior.

Model validation is fundamentally subjective. Modelers choose the validity tests, the criteria for passing those tests, what model outputs to validate, what setting to test in, what data to use, etc. Thus, the assertion "the model was judged valid" can mean almost anything (See Anderson; and House and Ball for elaboration). Nonetheless, a model validation effort will reveal model strengths and weaknesses which is valuable to users and those who extract information from model results.

Two validation approaches may be used: validation by construct and validation by results. Validation by construct asserts the model was built properly therefore it is valid. Validation by results refers to exercises where the model outputs are systematically compared against real world observations.

### **18.3 Validation by Construct**

Validation by construct is always used in modeling, but it is also the end of most of the programming model validation exercises. Validation by construct, as the sole method of validation, is justified by one of several assertions about modeling .

The right procedures were used by the model builder. Usually this involves the assertion that the approach is consistent with industry, previous research and/or theory; and that the data were specified using reasonable scientific estimation or accounting procedures (deducing the model data from real world observations).

Trial results indicate the model is behaving satisfactorily. This arises from a nominal examination of model results which indicates they do not contradict the modeler's, user's, and/or associated "experts" perceptions of reality.

Constraints were imposed which restrict the model to realistic solutions. Some exercises use constraints to limit adjustment possibilities and force the model to give results very close to historically observed outcomes. The application of "flexibility" constraints (Day and Cigno; Sahi and Craddock [1975, 1974]) as in the recursive programming example is such an approach.

The data were set up in a manner so that the real world outcome had to be replicated. In some models one can assure replication of a real world outcome through the model structure and data calculation procedures. This approach is manifest in input-output modeling (Leontief, 1936) where procedures insure that the base solution will always arise. A similar approach has appeared in price endogenous programming applications (see Miller and Millar; Fajardo et al.).

Fundamentally, validation by construct suffers from the shortcoming that validation of a particular model is assumed, not tested. If a model plays an integral part in a study, going forth with a model that is only assumed valid does not appear to be totally satisfying. However, validation by construct is a necessary precursor to any validation by results testing.

## 18.4 Validation by Results

Validation by results involves comparison of model solutions with real world outcomes. Models used in such a comparison will always have been built relying on experience, precedence, theory, appropriate data estimation and measurement procedures. Thus, validation by construct will always precede validation by results<sup>16</sup>. Testing whether the model output reasonably reproduces real world results is the next validation step. That determination involves five phases: first, a set of real world outcomes and the data causing that outcome is gathered; second, a validation experiment is selected; third, the model is set up with the appropriate data, the experiment is implemented and a solution is generated; fourth, the degree of association between model output and the real world outcome is tested; and, finally, a decision is made regarding model validity. Comments relative to these steps appear below.

### 18.4.1 Parameter Outcome Sets

Data describing a real world observation contain both the values for the model input parameters and model output measures. Thus, when predicting corn acreage one needs the prices, costs and resource endowments that led to the acreage decision. Tests of the model beyond the original data set will generally be more representative of model accuracy in applications (Anderson and Shannon). While complete input parameter-outcome sets are most desirable, partial sets giving aggregate measures (giving total corn acreage - not acreage by planting date) can be useful.

### 18.4.2 Validation Experiments

A set of validation experiments is described below. These experiments are not mutually exclusive; rather they are a set of sequential experiments which should be performed (or at least considered) in a given order. Five general validation experiments will be presented: a feasibility experiment, a quantity experiment, a price experiment, a prediction experiment, and a change experiment.

#### 18.4.2.1 Feasibility Experiment

The feasibility experiment has primal and dual forms. The basic idea involves setting up the model equations with the variables held at their observed levels, then examining solution feasibility.

The primal test involves addition of the constraints:

$$\begin{aligned} X' & \bar{X} \\ Y' & \bar{Y} \\ Z' & \bar{Z} \end{aligned}$$

This experiment tests internal model consistency. Often, the feasibility experiment is neglected in favor of,

---

<sup>16</sup> We do not recommend imposing constraints or model structure to force validation unless they are absolutely necessary and certainly not until initial testing has been done

for example, seeing if the model can replicate  $(\bar{X}, \bar{Y}, \bar{Z})$ . However such a solution can never be replicated if it is not feasible. The feasibility experiment often determines needed data, data calculation or model structure revisions. Such an experiment also finds errors arising due to faulty model equation specification.

The dual feasibility experiment involves testing whether the observed shadow prices are feasible in the dual or the Kuhn-Tucker conditions. For the example above this involves seeing whether:

$$\begin{array}{rcl}
 U & & \# \quad f'(X) \\
 \& \text{ UD} \quad \% \text{ VG} \quad \% \text{ WA} & \# \quad 0 \\
 & \& \text{ V} & \# \quad \& g'(Z) \\
 U & & ' \quad \bar{U} \\
 & \text{V} & ' \quad \bar{V} \\
 & & \text{W} \quad ' \quad \bar{W}
 \end{array}$$

is feasible<sup>17</sup>. This procedure tests whether the solution is dual feasible and therefore primal optimal. Non-zero variables in the observed outcome should, because of complementary slackness, lead to equality Kuhn-Tucker conditions. Zero variables should ordinarily be associated with strict inequalities. Careful execution of this experiment quite often reveals inadequacies in structure, data, or the objective function. Again, there is the attendant possibility of an inconsistent "real world outcome" which requires correction.

The data requirements of these feasibility conditions are rather strong -- they assume knowledge of a complete solution. Often, one may know output and input levels  $(X, Z)$  and aggregate sums of production variables (sums within  $Y$ ) but not individual variable values. Thus, tests involving totals may be in order. Second, the experiments may require artificial variables to both allow and help find infeasibilities as discussed in the last chapter.

#### 18.4.2.2 Quantity Experiment

The quantity experiment involves constraining the outputs supplied or inputs demanded at their actual levels and removing  $f(X)$  or  $g(Z)$  then observing the shadow prices. The output variant (developed by Kutcher) involves adding the constraint

$$X \leq \bar{X}$$

with the objective function term  $f(X)$  dropped. The correspondence of  $\bar{Y}, \bar{Z}, \bar{U}, \bar{V}, \bar{W}$  and the shadow prices on the equation pegging the  $X$  value can then be tested.

Such a test examines the consistency between the optimal and observed levels of the production ( $Y$ )

---

<sup>17</sup> Note the ' sign of  $f(x)$  denotes the first derivative.

and input supply ( $Z$ ). In addition, the imputed values of the resources ( $V$  and  $W$ ) may be examined for consistency with the observed values. Further, the dual values associated with the quantity constraints ( $X^* - \bar{X}$ ) should be sufficiently close to the market price for the outputs.

These shadow prices give an indication of the marginal cost of production at the observed quantities (Figure 18.1). This procedure is a test of the economic assumption of perfect competition (Kutcher) since, under perfect competition, the shadow price should equal the market price.

The input version of the test is essentially identical. The model is augmented by the constraint  $Z^* - \bar{Z}$  with the  $g(Z)$  term dropped from the objective function. In this case, the experiment should generate dual variables which can be compared to prevailing market prices of inputs (Figure 18.2).

#### 18.4.2.3 Price Experiment

A third type of model validation experiment is the price experiment. This type of experiment is relevant in price endogenous models or models with fixed demand requirements. This experiment involves fixing the objective function coefficients at existing real world prices ( $U, W$ ), then observing quantities (the dual of the quantity experiment). The optimal quantities ( $X^*, Z^*$ ) are then compared to the observed levels ( $X, Z$ ). The output price experiment is illustrated in Figure 18.3 where the fixed output price is equated with the model supply schedule to get a value of  $X^*$ . One may also examine how implicit fixed resource values are influenced in the experiment.

#### 18.4.2.4 Prediction Experiment

The prediction experiment is the most common validation by results test. Examples can be found in Barnett, et al. (1982); Brink and McCarl (1979); and Hazell and Pomareda. The prediction experiment involves fixing the problem data at real world values and solving to get  $X, Y, Z$ . In turn we test whether the linear programming model output is close enough to the real world outcomes.

#### 18.4.2.5 Change Experiment

The prediction experiment is to some degree the ultimate validation experiment in that it tests whether a model can replicate reality. However, most programming models are used for comparative statics analysis. This implies a need for an experiment which tests whether models accurately predict change.

To test a model's ability to predict change, one must have data on two real world situations and the resultant model solutions. Then, a comparison is made between the change in the model solution variables (e.g.,  $X_1^*, X_2^*$ ) and the change observed in the real world solution ( $\bar{X}_1, \bar{X}_2$ ) as done in Hazell et al. (1981).

#### 18.4.2.6 Tracking Experiment

Even a model which satisfactorily predicts a one time change may not be adequate. One may wish to track adjustments through time. For validation of such applications the model can be solved using a series of parameter sets. The focus of the validation would then be on how well the model "tracks" over



time with respect to the corresponding observed adjustments in the system. Again, comparisons are made between changes in the model solution and observed changes in the real world solution (for example, see Pieri et al.).

#### 18.4.2.7 Partial Tests

The above experiments are discussed for a model as a whole. Obviously, in any particular validation exercise, it may be desirable to perform experiments with some of the variables fixed at real world levels with other variables left unconstrained -- an attempt to validate portions of the model. Often, this type of experiment will be necessary with large models because observations on all decision variables and/or shadow prices may not be readily available. Validation experiments may then be performed to require sums of variables to equal observed real world levels, for example.

#### **18.4.3 Employing a Validation Test**

There are several identifiable stages for conducting one of the experiments given the model.

Step 1. Alter the model variables, equations and data to reflect the validation experiment.

Step 2. Solve the model(s).

Step 3. Evaluate the solution(s). Is it infeasible, unbounded, or optimal?

- (a) If the model solution is infeasible, examine the results to find the cause of infeasibility. Use the artificial variable based method in the last chapter. Once the cause is found, go to Step 5.
- (b) If the model is unbounded, use the large upper bound method from the last chapter. Once the cause is found, go to Step 6.
- (c) If the solution is optimal, perform association tests (as discussed below) to discover the degree of correspondence between the "real world" and the model solutions (except for the feasibility experiment). These tests should be conducted upon both the primal and dual variables.

Step 4. If the model variables exhibit a sufficient degree of association, then:

- (a) do higher level validation experiments, or
- (b) determine the model is valid and proceed to use it.

Step 5. If the model does not pass the validation tests, consider whether:

- (a) the data are consistent and correctly calculated,
- (b) the model structure provides an adequate representation of the real world system, and
- (c) the objective function is correctly specified.

Step 6. Fix the Model --Procedures for recalculating model parameters will be problem specific.

If, for example, all the variables have been fixed at "real world" levels and infeasibilities

occur, then the units of the observed input parameters and outputs may be inconsistent. If the data are accurate and model structure problems are suspected, one should consider whether: errors have been made in constructing the matrix; additional constraints are needed; or such factors as risk and/or willingness to adjust (i.e., flexibility constraints) should be entered into the model. If the model has been respecified either structurally or through its data, proceed back to Step 3 and repeat the validation test. If not, go to Step 7.

Step 7. If the preceding steps do not lead to a valid model, one must decide whether to:

- (a) do demonstrations with an invalid model -- assuming this is an approximately correct structure,
- (b) abandon the project, or
- (c) limit the scope of validation to a lesser set of variables (aiming at a less strict level of validation), subsequently qualifying model use. This may happen in many cases due to some considerations discussed subsequently.

#### **18.4.4 Evaluation Criteria for Validation**

Association tests can be used to measure whether a set of model results are similar to observed results. Quite a number of association tests are available as reviewed by Shannon; Anderson; Gass (1983); or Johnson and Rausser, for example. These tests have been well presented elsewhere and their theoretical roots are well outside our scope, so only a brief discussion will be given.

Regression techniques have been used to measure the association of model solutions with observed values (for examples see Nugent; Rodriguez and Kunkel). In that case, model results are regressed on observed values with perfect association indicated by an intercept of zero and a slope of one. The Theil U test has also been used (Leuthold; Pieri, et al.). This is a nonparametric "goodness of fit" test. Garret and Woodworth suggest the use of the G Index for validation -- a procedure for comparing sets of basic variables (an example can be found in Keith). Simple measures such as means, sums, mean absolute deviations, and correlation coefficients, have been used (Nugent; Kutcher; Hazell, et al., 1981). The authors have not found applications of Kolmogrow-Smirnov, Chi Squared or various other "goodness of fit" tests in a programming context. However, these techniques have been applied in simulation settings (see Anderson; Johnson and Rausser; Shannon; and Gass).

#### **18.5 What if the Model Does not Validate**

From a practical standpoint, models do not always pass validation tests. Since models always involve many assumptions, failure to validate, likely indicates that improper assumptions have been used. Consequently, when models fail validation tests, modelers often ask: What assumptions should be corrected?

As discussed above, programming models embody assumptions about both mathematical structure and the model structure. The mathematical structure assumptions involve additivity, divisibility, certainty, and proportionality. These assumptions, when severely violated, will cause validation tests to fail. The model designer then must consider whether these are the cause of failure. If so, the use of techniques such as separable, integer, nonlinear, or stochastic programming may be desirable to construct a new model.

Modeling assumptions may also lead to validation test failure. These assumptions embody the correctness of the objective function, variables, equations included, coefficients, and equation specification. Programming algorithms are quite useful in discovering assumption violations. Given an optimal solution, one may easily discover what resources were used, how they were used, and their marginal values. Thus, when presented with an invalid solution, resource usage and resource valuation should be investigated. Models are most often invalid because of inconsistent data, bad coefficient calculation, bad equation specification, or an incorrect objective function. Thus, common fixes for a model failing validation involve data respecification and/or structural corrections.

When dealing with linear programming, there are several other properties which can lead to validation failures. An optimal LP solution is characterized by the term basic, i.e., no more activities can be in the model than the number of constraints. For example, if a disaggregated regional model is constructed with a single constraint in each region, at most one activity will be produced in each region (if other constraints are not present in the model). This is ordinarily inconsistent with real world performance. Models then may be judged invalid because they overspecialize in production due to the nature of basic solutions. Several approaches may be taken when faced with this sort of inadequacy in a model solution. First, one may be satisfied with validating only aggregate results and not worrying about individual production results. Second, one may constrain the model to the observed solution and investigate whether this solution is an alternative optimal solution (which, as argued by Paris, may commonly occur). Third, one may recognize that a basic solution will not validate and enter constraints that limit the adjustment process of the activities within the model (flexibility constraints (Day) or aggregation procedures (Onal and McCarl [1991, 1989]) as discussed in the price endogenous chapter). Fourth, the model may be expanded by including risk considerations. Fifth, one may feel the model is structurally inadequate in that many of the factors that constrain production may be inadequately portrayed in the model (see the arguments in Baker and McCarl). Such a situation leads to either one of two fixes: more constraints can be added or the activities within the model may be respecified so they represent feasible solutions within omitted constraints as in the price endogenous chapter (Onal and McCarl [1991, 1989]).

Models may also fail validation because of the objective function. Specification of the constraints identifies the set of possible solutions, while the objective function determines the single optimal solution.

Thus, the objective function must be carefully specified and reviewed (with the dual feasibility test used if possible). Finally, the objective function may generate alternative optimal solutions, one of which is the desired solution (see Paris or Burton et al. for discussion).

Another phenomena may cause models to fail validation tests. Operations, quite often, are performed over several time periods. An annual model depicting operations of this type may well be invalid because it ignores initial conditions or does not recognize that parameter expectations may change over time. Thus, unless the model has initial conditions identical to those in the "real world," it may be very difficult to validate.

### **18.6 Comments**

Validation is an important concern within any programming exercise. A well validated model will have gone through both validation by construct and validation by results phases. Unfortunately, true validation will never occur as models can only be proved invalid. However, through satisfactory completion of the above experiments, the level of satisfaction may be increased.

The ultimate test of validity deals with adoption of the model by the decision maker. Satisfactory validation via the procedure given may not be sufficient for acceptance. A numerically valid model may solve the wrong problem and thus, will never be valid from the decision maker's viewpoint. Clearly, under these circumstances, validation in the broadest sense is only achievable by redefining the model so it takes into account the true problem.

## References

- Anderson, J. "Simulation: Methodology and Applications in Agricultural Economics." Review Marketing and Agricultural Economics, 43(1974):3-55.
- Barnett, D., B. Blake, and B. McCarl. "Goal Programming via Multidimensional Scaling Applied to Senegalese Subsistence Farms," American Journal of Agricultural Economics, 64,4(1982):720-7.
- Brink, L. and B. McCarl. "The Adequacy of a Crop Planning Model for Determining Income, Income Change, and Crop Mix." Canadian Journal of Agricultural Economics, 47(1979):13-25.
- Burton, R.O., J.S. Gidley, B.S. Baker and K.J. Reda-Wilson. "Nearly Optimal Linear Programming Solutions: Some Conceptual Issues and a Farm Management Application." American Journal of Agricultural Economics. 69(1987):813-818.
- Day, R. Recursive Programming and Production. North Holland Publishing Company, Amsterdam, 1963.
- Day, R. and A. Cigno, editors. Modeling Economic Change: The Recursive Programming Approach. Amsterdam: North-Holland Publishing Co., 1978.
- Fajardo, D., B. McCarl, and R. Thompson. "A Multicommodity Analysis of Trade Policy Effects: The Case of Nicaraguan Agriculture." American Journal of Agricultural Economics, 63,1(1981):23-31.
- Garret H. and R. Woodworth. Statistics in Psychology and Education. New York: David McKay Co., Inc., 1964.
- Gass, S.I. "Decision-Aiding Models: Validation, Assessment, and Related Issues for Policy Analysis." Operations Research, 31(1983):603-31.
- Hazell, P. and C. Pomareda. "Evaluating Price Stabilization Schemes with Mathematical Programming." American Journal of Agricultural Economics. 63,3(1981):550-6.
- Hazell P., R. Norton, M. Parthasarthy, and C. Pomareda. "The Importance of Risk in Agricultural Planning Models." Programming Studies for Mexican Agricultural Policy, R. Norton and L. Solis (eds.). New York: Johns Hopkins Press, 1981.
- Henderson, J. "The Utilization of Agricultural Land: A Theoretical and Empirical Inquiry." Review Economics and Statistics, 41(1959):242-59.
- Hillier, F. and G. Lieberman. Introduction to Operations Research. San Francisco: Holden-Day, Inc., 1967.
- House, P. and R. Ball, "Validation: A Modern Day Snipe Hunt: Conceptual Difficulties of Validating Models," in S.I. Gass (ed.), Validation and Assessment of Issues in Energy Models, Proceedings of a Workshop, National Bureau of Standards, Spec. Pub. 564, Washington, D.C.;1980.
- Johnson, S. and G. Rausser, "Systems Analysis and Simulation: A Survey of Applications in Agricultural and Resource Economics," in L. Martin (ed.), A Survey of Agricultural Economics Literature, 2(1977):157-301.
- Keith, N. "Aggregation in Large-Scale Distribution Systems." Ph.D. thesis, Purdue University, 1978.
- Kutcher, G., "A Regional Agriculture Planning Model for Mexico's Pacific Northwest." in R. Norton and L. Solis (eds.), Programming Studies for Mexican Agricultural Policy, Chapter 11, New York: Johns Hopkins Press, 1983.
- Leontief, W. "Quantitative Input-Output Relations in the Economic System of the United States," Review of Economic Studies, 18(1936):105-25.
- Leuthold, R.M., "On the Use of Theil's Inequality Coefficients." American Journal of Agricultural Economics, 57,3(1975):344-6.
- McCarl, B.A. "Model Validation: An Overview with Some Emphasis on Risk Models." Review of Marketing and Agricultural Economics. 52,3(1984):153-173.

- McCarl, B. "Cropping Activities in Agricultural Sector Models: A Methodological Proposal." American Journal of Agricultural Economics, 64,4(1982): 768-72.
- McCarl, B.A. and J.D. Apland. "Validation of Linear Programming Models." Southern Journal of Agricultural Economics. 68,5(1986):155-164.
- McCarl, B. and T. Spreen. "Price Endogenous Mathematical Programming as a Tool for Sector Analysis." American Journal of Agricultural Economics, 62,1(1980):87-102.
- Miller, T. and R. Millar, "A Prototype Quadratic Programming Model of the U.S. Food and Fiber System." mimeo, CES-ERS-USDA and Department of Economics, Colorado State University, 1976.
- Nugent, J., "Linear Programming Models for National Planning: Demonstration of a Testing Procedure." Econometrica. 38(1970):831-55.
- Onal, H. and B.A. McCarl. "Exact Aggregation in Mathematical Programming Sector Models." Canadian Journal of Agricultural Economics. 39(1991):319-334.
- Onal, H. and B.A. McCarl. "Aggregation of Heterogeneous Firms in Mathematical Programming Models." European Journal of Agricultural Economics. 16,4(1989):499-513.
- Paris, Q. "Multiple Optimal Solutions in Linear Programming Models." American Journal of Agricultural Economics. 63,4(1981):724-7.
- Pieri, R., K. Meilke, and T. MacAuley, "North American-Japanese Pork Trade: An Application of Quadratic Programming." Canadian Journal of Agricultural Economics, 25(1977):61-79.
- Rodriguez, G. and D. Kunkel, "Estimation of Flexibility Coefficients for Recursive Programming Models--Alternative Approaches." American Journal of Agricultural Economics, 56,2(1974):344-50.
- Sahi, R. and W. Craddock. "Estimating Crop Acreages in the Prairie Provinces: Application of Recursive Programming." Canadian Journal of Agricultural Economics. 23(1975):1-16.
- \_\_\_\_\_. "Estimation of Flexibility Coefficients for Recursive Programming Models - Alternative Approaches." American Journal of Agricultural Economics. 56(1974):344-50.
- Shannon, R. "Simulation: A Survey with Research Suggestions." AIIE Transactions, 7(1975):289-301.

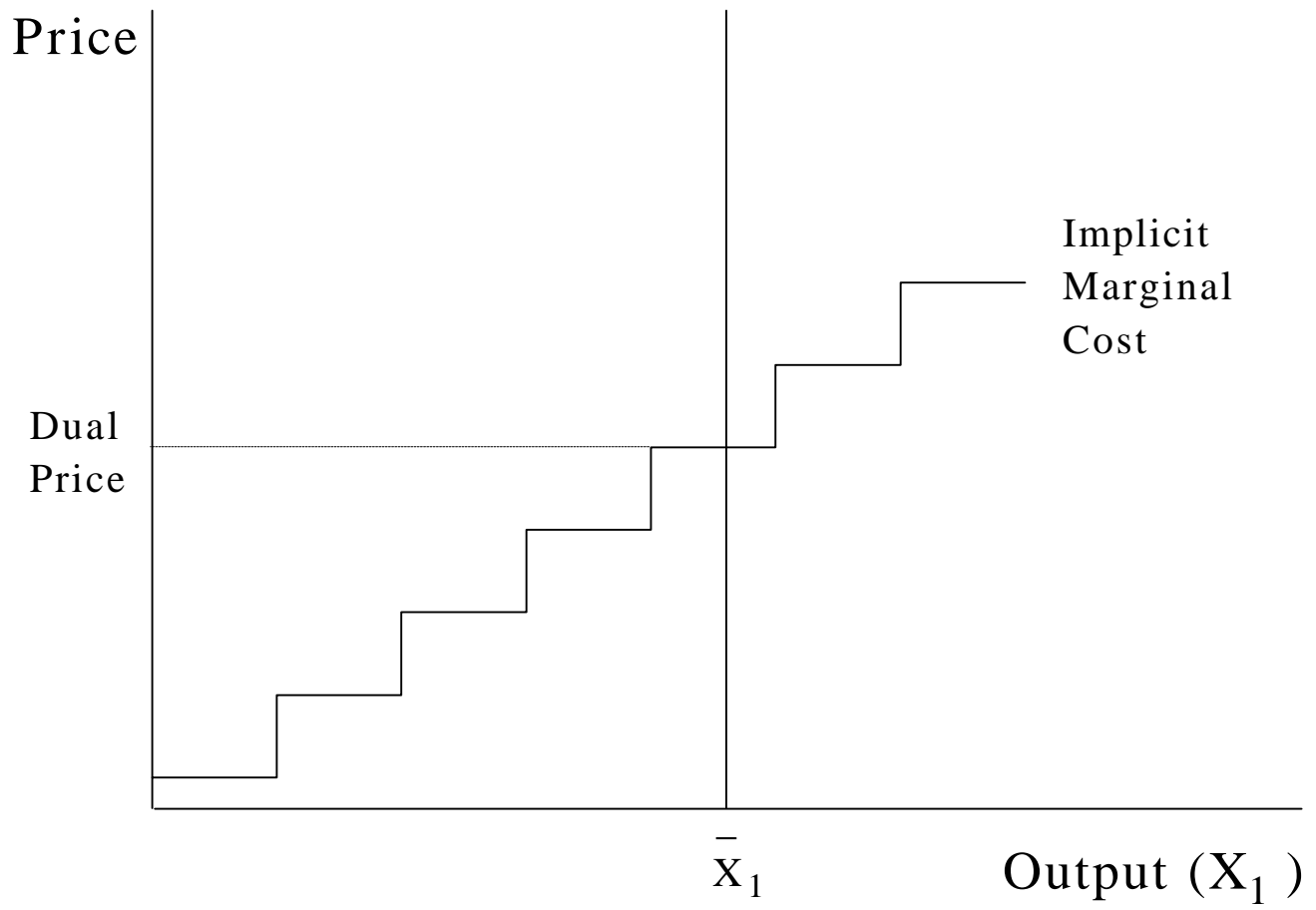


Figure 18.1 Output Quantity Experiment

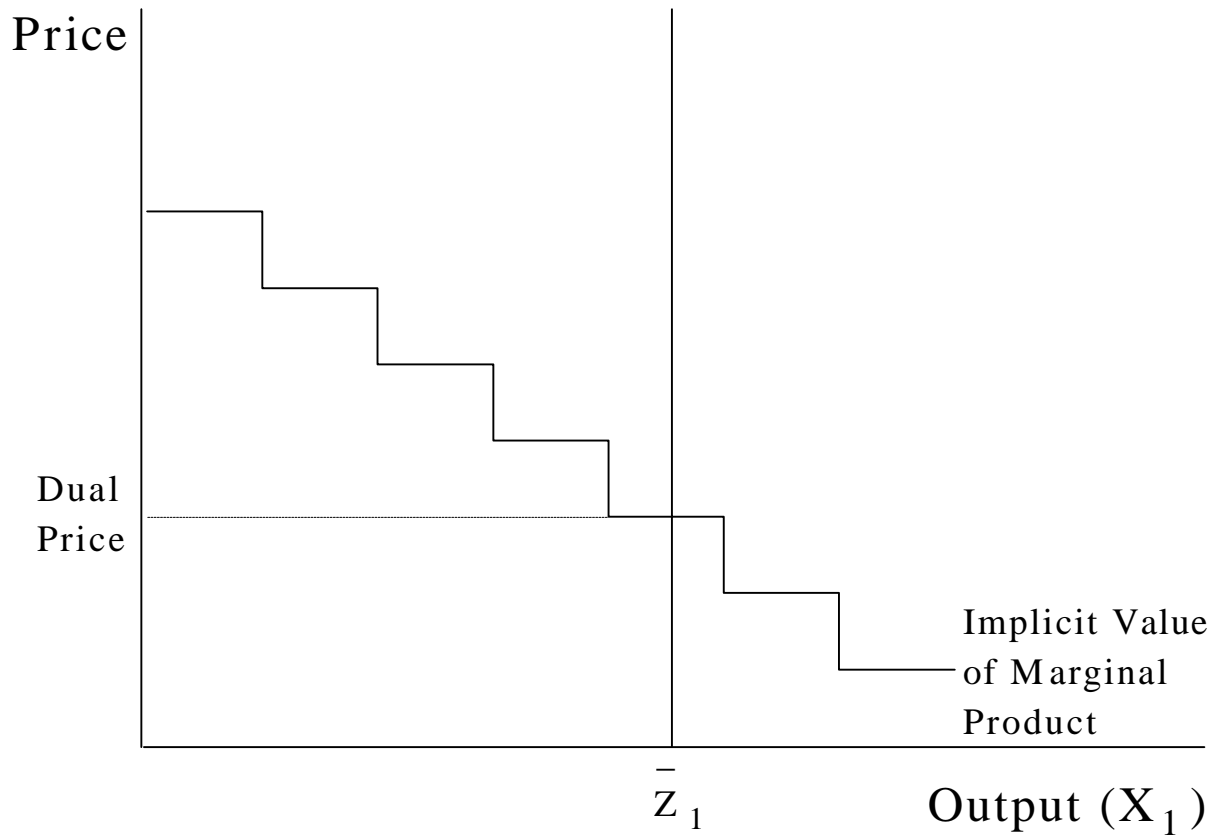


Figure 18.2 Input Quantity Experiment



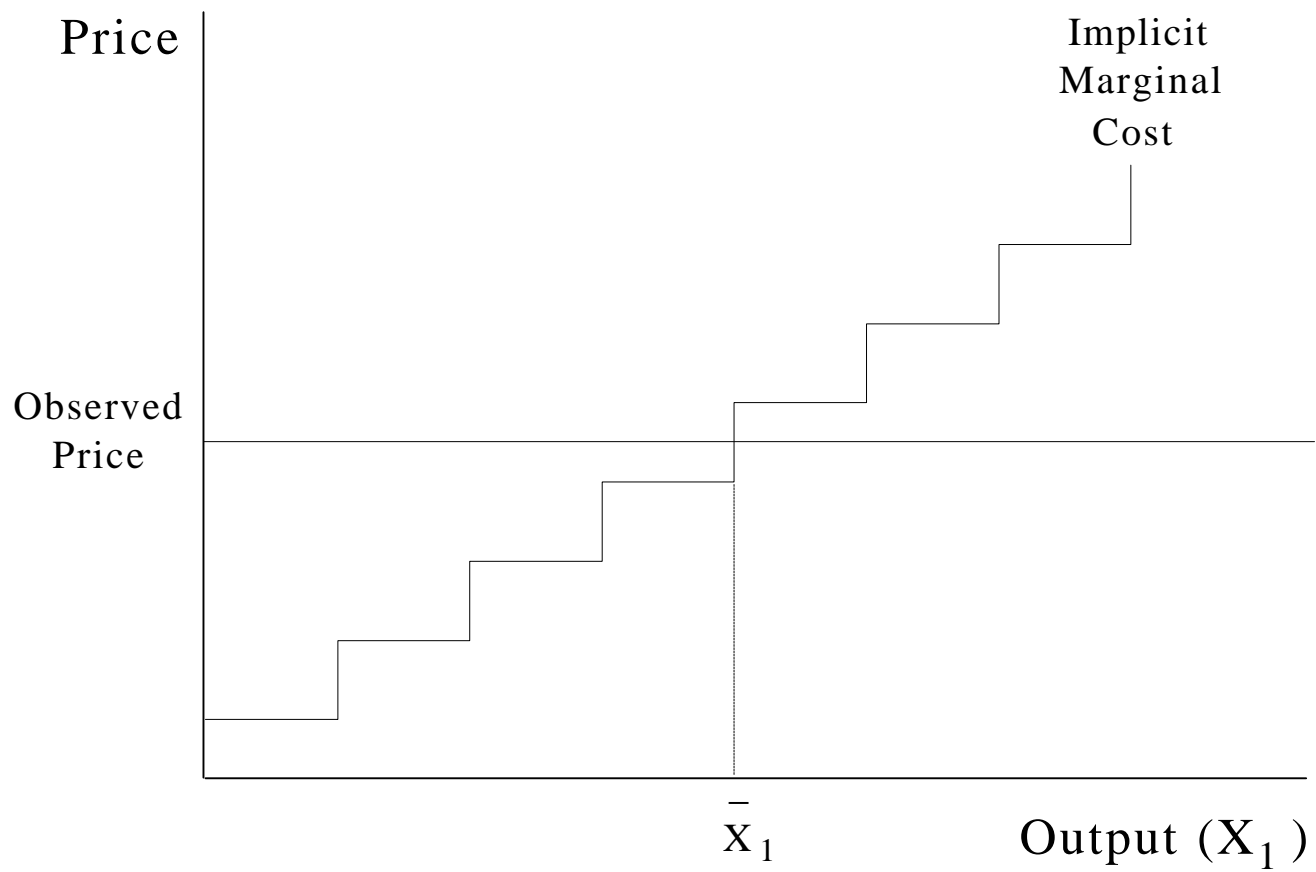


Figure 18.3 Output Price Experiment

## CHAPTER XIX: COMPUTERIZED AIDS FOR PROGRAMMING STUDIES

Data manipulation, model specification, debugging, validation and experimentation within moderate sized programming models can be complex. Computerized procedures may facilitate these tasks. This chapter reviews ways computerized procedures may facilitate programming studies. In addition, material on modeling with GAMS will be presented.

Given the model structure and data, computerized procedures may be used to retrieve necessary data items; calculate coefficients; prepare equations for solver input; check numerical formulations for errors; summarize solution content; debug the model; predict implications of data alterations; and execute model validation tests. Many of these options are available within modeling systems such as GAMS, but some are not and a more general discussion is in order.

Before discussing computerized approaches, three caveats are pertinent:

- 1) This chapter assumes that a programming formulation is the appropriate analysis tool.
- 2) The chapter is based upon the authors' experience and does not reflect experience with packages such as OSL or ANALYZE (Greenberg, 1991).
- 3) The chapter is oriented toward solving moderate to large models (more than 100 rows and/or columns).

### 19.1 Model Generation Tools

Numerical programming models must be constructed in a form consistent with solver input requirements. Construction of the numeric formulation is commonly called matrix generation. Problems of the size dealt with herein virtually require a computerized matrix generator (MG). Williams (1978a, pg. 35) argues that "the clerical problems (of building moderate to large models by hand) almost always become prohibitive."

Different types of MGs can be employed. These run the continuum from simple specific model MGs to MGs designed to facilitate direct model use by decision makers (see McCarl and Nuthall or McCarl, et al.) on to general purpose matrix generators as in modeling systems (i.e., GAMS-Brooke, et al.). The fundamental distinction made here is between custom, problem specific and general purpose modeling system based MGs.

#### 19.1.1 Potential Functions of a Matrix Generator

Potentially there are several functions MGs can perform. We group these into input preparation, formulation manipulation, documentation and analysis facilitation.

##### 19.1.1.1 Input Preparation

Fundamentally MGs prepare solver input from data and equation specifications. Programming models usually permit a large model to be generated from a smaller set of input (e.g., Kutcher and Meeraus state that 400 lines of data and a 1000 statement FORTRAN matrix generator yielded an 80,000 line numerical model file). Matrix generators calculate and place coefficients avoiding clerical errors.

### 19.1.1.2 Formulation Manipulation

Initial formulations are almost always different from final formulations due to omissions, errors, improper data or oversights. Model based analysis usually involves new and altered equations and variables, as well as data changes. MGs enable rapid, efficient modification of model structure and/or data usually requiring modification of a few commands or input tables.

### 19.1.1.3 Documentation

Often, it is very hard for someone other than the model author to use a model. Documentation is often inadequate. MG use can partially solve the documentation problem. MGs utilize specific data, carry out data transformations, and place the coefficients. This contributes to documentation. Nevertheless, accompanying documentation is still important.

### 19.1.1.4 Analysis Facilitation

MGs may facilitate analysis reducing the time to do a run and correct errors. Thus, a more extensive analysis may be done (since it is easier to do individual runs). Other possible MG functions include: suggesting an advanced basis (Dillon); resolving degeneracies (McCarl, 1977); resolving alternative optimals (McCarl et al., 1977); checking for inconsistent or improper data -- when the modeler codes in such checks; aiding in the discovery of causes of infeasibility (adding artificial variables) or unboundedness (by adding large upper bounds); and automatically setting up approximations of nonlinear phenomena (i.e., generating a separable programming approximation).

## **19.1.2 When not to use a Matrix Generator**

MGs are not always appropriate. MGs should only be used when the time required to do all analysis with the MG is less than or equal to the time to do the analysis without it. (This is an elusive criterion since many uncertain factors enter.) Thus, if the analysis may be done efficiently without a MG, don't use one. This criterion also applies to the choice between a generalized (e.g., GAMS) and a problem tailored MG. Problem specific MGs are more time consuming to construct and can be difficult to change because of accompanying documentation and complexity of the implementation. This situation could well lead to an analysis done with the wrong model. A matrix generator should be used with the attitude that it will change to accommodate the problem rather than vice versa.

## **19.2 Pre-Solution Formulation Analysis Tools**

The first problem formulation is rarely the final formulation used in the study. Computerized tools may facilitate the identification of formulation difficulties. Unfortunately, GAMS does not contain much in the way of such tools although GAMSCHK (McCarl 1994) does contain such tools. The primary aid in verification is a "picture" routine which generates a schematic of a programming problem (Orchard-Hays). A page of "picture"

output (e.g., Figure 19.1) typically represents 50-60 columns and 40-50 rows. Characters on the page indicate the location, sign and magnitude of a coefficient. A picture thereby allows verification of whether coefficients with proper sign and magnitude are in their proper places.

Preanalysis routines may also generate data on problem scaling, and check for obvious structural defects. Finally, presolution formulation analysis may use redundancy identification techniques (Luenberger, Ch. 6) which are an integral part of some large commercial packages (e.g., OSL automatically reduces the problem).

### **19.3 Solution Interpretation Aids**

Computerized techniques can be developed to aid solution interpretation. They can be used to summarize solutions, aid in solution debugging, or augment solution information.

#### **19.3.1 Solution Summarization**

The primary solution summarization tool is the report writer (RW). A RW takes solution information and reformats it into tables for decision makers.<sup>18</sup> Linear programming solver output is usually an inadequate description of a solution. Substantive summary reports are usually necessary and may be the only effective way to involve nontechnical decision makers in model use. Report writers are frequently based upon rudimentary accounting procedures involving data tabulation, labeling and organization. (McCarl and Nuthall discuss report writers at length.)

RWs may be either problem specific or general purpose. The arguments for a problem specific report writer are weaker than those for a matrix generator. This is due to both the ability to add accounting equations summing up variables of interest and the report writing features of modeling systems like GAMS (see the discussion of GAMS report writing later in this chapter). The proper trade-off in selecting whether to develop a problem specific report writer is between time required to do the whole analysis with the report writer versus other methods (hand and/or generalized report writers).

#### **19.3.2 Solution Debugging**

Post-solution computational aids may also help in debugging "bad" solutions. Often, the analyst has doubts about the validity of an LP solution. Specifically, three potential questions are asked: 1) Why is a particular variable in (or not in) the solution?; 2) Why is a particular shadow price so large?; and 3) Why does a particular row exhibit slack resources? The resolution of these questions revolves around the budgeting and row summing procedures discussed above. These items are easily computerized and have been in GAMSCHK (McCarl 1994).

---

<sup>18</sup> For example, see the GAMS report writing example above, or McCarl et al.

### 19.3.3 Solution Interpretation

Post solution computational aids can enhance or augment solution information. Linear programming solver output typically consists of solution levels, shadow prices, reduced costs and slacks. Less often, cost and right hand side ranges are obtained. Post optimality routines (or programs which request parametric analysis) may provide additional information (e.g., the impact of ranging a number of right hand sides simultaneously). Similarly, calculations may be done on, or requests generated for, information from the final tableau. (In larger problems this information is virtually never available and may be difficult to calculate. Although, it is sometimes quite useful.)

## 19.4 Getting the most from GAMS

There are a number of GAMS features which can facilitate a programming exercise. These features will be discussed under the topics: setting up data, changing model structure, providing an advanced basis, report writing, debugging models, conducting a comparative model analysis, sensitivity analysis, speeding up GAMS, minimizing model size, and avoiding GAMS failures.

### 19.4.1 Setting Up Data

GAMS has excellent facilities for data entry and computation as well as the ability to document data. We recommend that users enter primary data with comments on data sources and long labels. In turn, one should then calculate the needed data. External calculations are usually inadvisable as they are difficult to trace at a future time. For example, in the transportation example the distance and the formula relating distance to cost were entered with the costs computed. Users following such a strategy would find it easy to alter raw data assumptions and reanalyze the problem.

### 19.4.2 Changing Model Structure

One of the big advantages of using a modeling system is the ability to add constraints or variables and reanalyze the problem. However with large models or comparative studies, it may be desirable to make equations or terms in equations temporarily active or inactive. This can be achieved with \$ controls. For example if the following lines were put in a GAMS problem

```
SCALAR    ISITACTIVE    tells whether items are active /0/
CONDEQ$ISITACTIVE..    sum(stuff,x(stuff)) =L= 1;
EQNOTH(index)..        sum(stuff,r(index,stuff)*x(stuff)) +
                        4*sum(stuff,Y(stuff))$ISITACTIVE =L= 50;
```

they would cause the CONDEQ equation and the Y term in the EQNOTH equations to only appear in the empirical model when the ISITACTIVE parameter was nonzero. Thus, the sequence

```
ISITACTIVE = 0;
SOLVE MODELNAME USING LP MAXIMIZING OBJ;
ISITACTIVE = 1;
SOLVE MODELNAME USING LP MAXIMIZING OBJ;
```

would cause the model to be solved with and without the constraint and term.

### 19.4.3 Providing an Advanced Basis

The importance of an advanced basis has long been recognized in mathematical programming. Models which take more than 24 hours to solve from scratch, can be solved in less than 3 hours from a good basis. GAMS can accept an advanced basis under certain conditions. In any model involving a sequence of solves, GAMS automatically uses the basis from the first solve to restart the second and later solves. But this does not carry over to models solved from scratch.

So what can you do when solving a model from scratch? Three possibilities are available. 1.

One can let GAMS solve without worrying with the basis. This should certainly be done for small models.

2. One can try to suggest a basis by setting the levels and marginals for the variables and the marginals for the equations. For example the statements

```
X.L("VAR1") = 10;  
X.L("VAR2") = 0;  
X.M("VAR1") = 0;  
X.M("VAR2") = 10;  
CONSTRAINT.M("RES1")=20;  
CONSTRAINT.M("RES2")=0;
```

would result in a basis being suggested with the first variable and the slack from the second constraint.

3. One can save the variable and equation information from previous solves and restart including that information by setting the levels and marginals as above (a small example is given in the context of the Chapter 5 DIET problem in the files SAVBASIS and ADVBASIS) GAMSBAS (McCarl 1994a) does this automatically.

### 19.4.4 Report Writing

GAMS report writing has been discussed before in this text. However, a few additional comments are relevant. The normal GAMS output format is not always desirable. All \*.lst files have the GAMS instructions listed first and may be followed by: a cross reference map; output from pre-solution displays; the equation and variable listings; the solver output; and post-solution displays. Several methods may be employed to manage this output. One may

1. Suppress the cross reference map by using the command

```
$OFFSYMLIST OFFSYMXREF
```

2. Suppress the variable and equation listings by using

```
OPTION LIMROW = 0;  
OPTION LIMCOL = 0;
```

3. Limit the amount of GAMS code echoed back by restarting the job with a very short file containing

display statements only. For example the batch file (REDUCED.BAT)

```
GAMS MAINPART S=SAVED
GAMS FINALOUT R=SAVED
```

does this where MAINPART contains all the model statements ( in the example on the disc these are from the file EVPORTFO) and FINALOUT is just a display of the relevant output. This generates a much smaller FINALOUT.LST file.

4. Suppress the solution information using

```
OPTION SOLPRINT=OFF;
```

5. Suppress the listing of lines between \$OFFLISTING and \$ONLISTING commands

One may also wish to control the GAMS output ordering and format. In this context one should be aware that

- a. The order in which things appear in terms of elements of sets, variables etc. is determined by the order in which GAMS saw those items. In particular the order that the variables appear in the equation, variable and solution listings depends on the ordering of the VARIABLES declarations. Thus, if you want the TRANSPORT variable to proceed the DEMAND and OBJ variables you need to declare them in that order. The same holds for the equations.

The ordering of set elements is somewhat more tricky and depends on the order in which the set elements were seen across all sets. Thus the set statements

```
SETS ONE / SALT, PROTEIN , TOTAL/
      TWO / INDUSTRIAL , MUNICIPAL, TOTAL/
```

would result in any listing over data indexed by the set TWO appearing with the TOTAL element first since that word appeared first in set definitions before the other words.

The ordering of items in a parameter defined over multiple sets is also an issue. GAMS output is ordered with the sets varied from left to right. Thus, when displaying X(A,B,C) A will vary the slowest. The sets should be ordered in the pattern desired.

- b. The page width and page length are at the users control. Both can be specified in the GAMS initiation command. For example, the command

```
GAMS FILE PW=130 PS=9999
```

would result in 130 character wide output arrayed in 9,999 line long pages.

- c. One can control the formatting of output from a display statement using two option commands. The command

```
OPTION DECIMALS = 0;
```

would result in all displayed numbers being rounded to zero decimals (although GAMS still insists

on reporting all decimals for numbers less than one).<sup>19</sup> The command

```
OPTION TRADE:3:2:1;DISPLAY "TRADE FLOWS", TRADE;
```

would result in the data in the TRADE array being displayed with three indices displayed in the rows, two in the column headings, and the data would appear with one decimal place. Further, the label TRADE FLOWS would precede the display in the output.

- d. Users may create much more customized output files using the GAMS "PUT" command. This is a rather extensive feature (GAMS Development Corporation, 1992) but can generate output such as in Table 19.1 which constitutes a reformatting of the Report Writer output as tabled in chapter 6 (this is generated by the file PUTEXPL)

### 19.4.5 Debugging Models

GAMS allows one to work with large models. PC based GAMS can be used to solve problems with hundreds of thousands of variables and tens of thousands of equations. However, debugging formulations that big is not easy. The debugging aids in most commercial systems are not available within GAMS.<sup>20</sup> However, the algebraic structure of GAMS does provide one with the ability to **work from small to large**. This should always be exploited. For example, the full structure of a GAMS transportation problem implementation for a problem with N destinations and M sources can be worked out on a 2x2 problem. Data calculations, model equations, report writing tables, comparative model analysis procedures etc. can be worked out in a fast turn around limited output situation. Later the full data set can be entered.

GAMS supports several other debugging techniques:

1. One may examine the individual rows or equations within the model using the option statements for variable and equation listing. For example, the commands:

```
OPTION LIMROW=10;  
OPTION LIMCOL=10;
```

would result in the print out of the first 10 elements in each equation and variable block whenever a SOLVE is executed. One may control the ordering of these items so the proper items appear first by ordering the set elements and variable/equation names as discussed in the output control section above.

2. One may hand generate versions of the tests in the FIXING MODELS chapter using GAMS calculations. For example, in a resource allocation model the lines below could be used to test for

---

<sup>19</sup> One can overcome this using the command OUTPUTITEM\$(OUTPUTITEM GE 0.5) =OUTPUTITEM before the display statement.

<sup>20</sup> Moves are now afoot to correct that situation with the development of a GAMS interfaced version of ANALYZE (Greenberg, 1991) and GAMSCHECK (McCarl, 1994).



unbounded variables.

```
PARAMETER UNBOUN(J) equals 1 if variable j is unbounded;
UNBOUN(J)=1$(C(J) GT 0 AND SUM(I,1$(A(I,J) GT 0) LE 0);
DISPLAY UNBOUN;
```

Similarly, after solution, the following code would provide a budget analysis

```
PARAMETER BUDGET(J,*,*) BUDGET OF COLUMN J;
BUDGET(J,I,"AIJ")=A(I,J);
BUDGET(J,I,"SHADOWPRIC")=CONSTRAINT.M(I);
BUDGET(J,I,"PRODUCT")=A(I,J)*CONSTRAINT.M(I);
BUDGET(J,"SUMINDIRCT","PRODUCT")=
SUM(I,BUDGET(J,I,"PRODUCT"));
BUDGET(J,"OBJECTIVE","PRODUCT")=C(J);
BUDGET(J,"REDUCECOST","PRODUCT")=
BUDGET(J,"SUMINDIRCT","PRODUCT")-
BUDGET(J,"OBJECTIVE","PRODUCT");
DISPLAY BUDGET;
```

However, such coding is specific to a structure and is easier done using auxiliary aids such as those in McCarl (1994b); and Greenberg.

#### 19.4.6 Conducting a Comparative Model Analysis

Models, once setup, are usually employed in a comparative statics analysis. Such an exercise involves repeated solutions of the same problem. There are several GAMS features which are relevant in such a setting.

1. More than one model can be solved in a run. Thus, one can stack solve statements or loop over solves as in the LP modeling DIET and risk EVPORTFO example models.
2. When solving multiple model versions one needs to be careful with data revisions as the values in the model once changed remain so. For example the commands

```
SCALAR LAND /100/
PARAMETER SAVE LAND;
SAVE LAND = LAND;
SET LANDCHANGE SCENARIOS FOR CHANGES IN LAND
/R1,R2,R3/
PARAMETER VALUE(LANDCHANGE) PERCENT CHANGE IN LAND
/R1 10 , R2 20 , R3 30/
LOOP ( LANDCHANGE,
LAND = LAND * (1 + VALUE ( LANDCHANGE ) / 100. ) );
```

results in land equalling 110, 132 and 171.6 during the loop. However, alteration of the calculation statement, so it operated from a saved parameter value

```
LAND = SAVE LAND * (1 + VALUE ( LANDCHANGE ) / 100. )
```

result in values of 110 , 120, and 130.

3. The development of a comparative report writer may be attractive when doing multiple runs. Such a report writer is illustrated in the Risk chapter EVPORTFO example and is in TABLE 19.2. In that case a parameter is defined over the loop set - OUTPUT(\*,RAPS). In turn, during loop execution the OUTPUT array is saved with scenario dependent values of variables, shadow prices, data etc. Finally, when the output is displayed a comparison across scenarios appears.

### 19.4.7 Sensitivity Analysis

A number of users are interested in getting sensitivity analysis information from GAMS usually in the form of LP ranging analysis results. Unfortunately, the base version of GAMS does not yield such information. The user wishing for such information has two alternatives. First, one may cause the model to be repeatedly solved varying a parameter and examine the results (as in Table 19.2). Second, one can use solver dependent features of GAMS (which currently work with OSL or CPLEX) and retrieve the ranging information (GAMS Development Corporation, 1993)

### 19.4.8 Speeding up GAMS

One can speed up GAMS execution time. Models with a lot of subscripts and a lot of dimensions to those subscripts can be quite slow in performance. The use of \$ conditions in such models is essential. For example the report writing equation

$$Y = \text{SUM}((A, B, C, D, E, F, G), (\text{DAT}(A) + \text{IT}(B, C) + Y(D, E) + W(F, G)) * X.L(A, B, C, D, E, F, G))$$

will perform much faster with the addition of a \$ condition as follows

$$Y = \text{SUM}((A, B, C, D, E, F, G) \$ X.L(A, B, C, D, E, F, G), (\text{DAT}(A) + \text{IT}(B, C) + Y(D, E) + W(F, G)) * X.L(A, B, C, D, E, F, G));$$

this will result in the calculation only being done when nonzero values are involved and will avoid excess work.<sup>21</sup>

The ordering of subscripts is also important where the data arrays should be referenced in an order consistent with their definition. For example, summing the above in the order F,D,A,C,E,B,G would be much slower. Also one should compute intermediate products to avoid repetitive and complex calculations (i.e., one could add the DAT and IT items into another parameter ahead of time if they were frequently added). GAMS also gives help in reporting particularly slow statements. During execution, a report appears on the screen giving the line being executed, and one can observe progress making notes of statements which are computed for a long time to see if they can be streamlined. Also one may use the undocumented PROFILE feature which produces a report of the time spent on code segments<sup>22</sup>.

### 19.4.9 Minimizing Model Size

GAMS can generate very large problems when models contain a lot of sets with many elements. The statements above on speeding up GAMS are also relevant when setting up models. It is usually highly desirable to define \$ conditions on equations and the sums leading to generation of variables to avoid unneeded model features.

---

<sup>21</sup> Such a modification reduced run time for a report writer from 2 hours to 10 minutes.

<sup>22</sup> This is invoked by including the OPTION PROFILE=1, or including PROFILE=1 on the GAMS call.

#### **19.4.10 Avoiding GAMS Failures**

Finally we need to mention strategies for avoiding GAMS solver failures. Solver failures generally happen because of numerical difficulties caused by scaling, degeneracy, or nonlinearities. Cycling without making progress can often be resolved by adding small numbers to the right hand sides and/or scaling. Both of these procedures are explained in the fixing models chapter. Gradient problems or a lack of progress may require separable approximation of nonlinear problems as explained in the LP approximation chapter. Improved starting values for the nonlinear variables can also help.

## References

- Brooke, A., D. Kendrick, and A. Meeraus. GAMS: A User's Guide. Boyd and Fraser Publishers, Version 2.25, 1993.
- Dillon, M., "Heuristic Selection of Advanced Bases for a Class of Linear Programming Models," Operations Research, 18(1970):90-100.
- GAMS Development Corporation. "Sensitivity Analysis with GAMS/CPLEX and GAMS/OSL." Washington, DC, 1993.
- GAMS Development Corporation. "Guide to the 'Put' Writing Facility." Washington, DC, 1990.
- Greenberg, H.J. "A Primer for ANALYZE<sup>(c)</sup>: A Computer-Assisted Analysis System for Mathematical Programming Models and Solutions." Mathematics Department, University of Colorado at Denver, Denver, CO, February 1991.
- Kutcher, G. and A. Meeraus, "Computational Considerations for Sectoral Programming Models," in The Book of CHAC: Programming Studies for Mexican Agriculture. R. Norton and L. Solis, (Eds.), Johns Hopkins Press for the World Bank, 1983.
- Luenberger, D., Introduction to Linear and Nonlinear Programming, Addison Wesley, 1973.
- McCarl, B.A. "GAMSCHECK USER DOCUMENTATION: A System for Examining the Structure and Solution Properties of Linear Programming Problems Solved using GAMS." Working Documentation, Department of Agricultural Economics, Texas A&M University, 1994.
- McCarl, B.A. "Degeneracy, Duality and Shadow prices in Linear Programming," Canadian Journal of Agricultural Economics, 25(1977):70-73.
- McCarl, B.A., W.V. Candler, D.H. Doster and P. Robbins, "Experiences with Farmer Oriented Linear Programming for Crop Planning," Canadian Journal of Agricultural Economics, 25(1977):17-30.
- McCarl, B.A. and P. Nuthall, "Linear Programming for Repeated use in the Analysis of Agricultural Systems," Agricultural Systems, 8(1982):17-39.
- Optimization Subroutine Library (OSL). IBM Guide and Reference, Release 2. Ed. J.A. George and J.W.H. Liu. IBM Corporation, 1990-91.
- Orchard-Hays, W., Advanced Linear-Programming Computing Techniques, McGraw-Hill Book Company, 1968.
- Williams, H., Model Building in Mathematical Programming, John Wiley & Sons, 1978.

**Figure 19.1** Sample Picture of BLOCKDIAG Problem

GAMSCHECK PICTURE - COEFFICIENT CODES

LOWER BOUND (INCLUSIVE)	CODE	UPPER BOUND (LESS THAN)
1000.00000	G	+INFINITY
100.00000	F	1000.00000
10.00000	E	100.00000
1.00000	D	10.00000
1.00000	C	1.00000
.50000	B	1.00000
.00000	A	.50000
.00000	0	.00000
-.50000	1	.00000
-1.00000	2	-.50000
-1.00000	3	-1.00000
-10.00000	4	-1.00000
-100.00000	5	-10.00000
-1000.00000	6	-100.00000
-INFINITY	7	-1000.00000

```

|           M M M M M M           N   H   O I   E I   O
|           A A A A A A T T T T T T           E   S   S J   G J   W
|           K K K K K K R R R R R R           T           I ,   A ,
|           E E E E E E N N N N N N           I   C   T S   T S   C
|           T T T T T T S S S S S S           N   O   I   I   N
| M A K E C H A I R M A K A A A A A A P P P P P P S E L L S E L L S E C   E   V   V   T
|           B B B B B B O O O O O O           O   F   E   E   S
|           L L L L L L R R R R R R           M   F
|           E E E E E E T T T T T T           E   S
|           1 1 1           I
| 1 2 3 4 5 6 7 8 9 0 1 2 1 2 3 4 5 6 1 2 3 4 5 6 1 2 3 4 5 6 7 8 9 0 1
-----
OBJT 1| E E E E E E E E E E E E E E F   E F D D E E D D 6 6 6 7 5 6 6 6 5 6 C   = 0   2 3   1 0   3 3
R    1|           D D           < F   2   0   2
E    2|           C C           < E   2   0   2
S    3| B D A D D B           < F   6   0   6
O    4| B A D B A D           < E   6   0   6
U    5| A A A C C C           < F   6   0   6
R    6| C D D B B B           D D           < F   8   0   8
E    7|           C C           < 0   2   0   2
Q    8|           B D A D D B           < F   6   0   6
R    9|           B A D B A D           < F   6   0   6
E   10|           A A A C C C           < F   6   0   6
S   11|           C D D B B B           D D           < F   8   0   8
O   12|           C C           < E   2   0   2
LINKTABLE 1|           3           3           C   C           < 0   2   2   4
LINKTABLE 2|           3           3           C   C           < 0   2   2   4
LINKCHAIR 1|           3           3           D           < 0   1   2   3
LINKCHAIR 2|           3           3           D           < 0   1   2   3
TRNCHAIREQ 1| 3 3 3           C           C           < 0   2   3   5
TRNCHAIREQ 2|           3 3 3           C           C           < 0   2   3   5
TRNCHAIREQ 3|           3 3 3           C           C           < 0   2   3   5
TRNCHAIREQ 4|           3 3 3           C           C           < 0   2   3   5
TRNTABLEEQ 1|           3           C           C           < 0   2   1   3
TRNTABLEEQ 2|           3           C           C           < 0   2   1   3
-----
POSITIVE | 5 5 5 5 5 5 3 2 3 2 2 2 1 2 1 1 1 1
COLUMN CTS | 5 5 5 5 5 5 3 2 3 2 2 2 1 2 1 1 1
NEGATIVE | 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0
COLUMN CTS | 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1
COLUMN | 6 6 6 6 6 6 4 2 4 3 3 3 2 3 2 2 2 1
COUNTS | 6 6 6 6 6 6 4 2 4 3 3 3 2 3 2 2 2
-----

```

**Figure 19.1.** Sample Picture of BLOCKDIAG Problem(continued)

```
#### Dictionary of Variables
M 1: MAKECHAIR(PLANT2,FUNCTIONAL,NORMAL)
A 2: MAKECHAIR(PLANT2,FUNCTIONAL,MAXSML)
K 3: MAKECHAIR(PLANT2,FUNCTIONAL,MAXLRG)
E 4: MAKECHAIR(PLANT2,FANCY,NORMAL)
C 5: MAKECHAIR(PLANT2,FANCY,MAXSML)
H 6: MAKECHAIR(PLANT2,FANCY,MAXLRG)
A 7: MAKECHAIR(PLANT3,FUNCTIONAL,NORMAL)
I 8: MAKECHAIR(PLANT3,FUNCTIONAL,MAXSML)
R 9: MAKECHAIR(PLANT3,FUNCTIONAL,MAXLRG)
M 10: MAKECHAIR(PLANT3,FANCY,NORMAL)
A 11: MAKECHAIR(PLANT3,FANCY,MAXSML)
K 12: MAKECHAIR(PLANT3,FANCY,MAXLRG)
MAKETABLE 1: MAKETABLE(PLANT1,FUNCTIONAL)
MAKETABLE 2: MAKETABLE(PLANT1,FANCY)
MAKETABLE 3: MAKETABLE(PLANT2,FUNCTIONAL)
MAKETABLE 4: MAKETABLE(PLANT2,FANCY)
MAKETABLE 5: MAKETABLE(PLANT3,FUNCTIONAL)
MAKETABLE 6: MAKETABLE(PLANT3,FANCY)
TRNSPORT 1: TRNSPORT(PLANT2,CHAIRS,FUNCTIONAL)
TRNSPORT 2: TRNSPORT(PLANT2,CHAIRS,FANCY)
TRNSPORT 3: TRNSPORT(PLANT3,TABLES,FUNCTIONAL)
TRNSPORT 4: TRNSPORT(PLANT3,TABLES,FANCY)
TRNSPORT 5: TRNSPORT(PLANT3,CHAIRS,FUNCTIONAL)
TRNSPORT 6: TRNSPORT(PLANT3,CHAIRS,FANCY)
S 1: SELL(PLANT1,TABLES,FUNCTIONAL)
E 2: SELL(PLANT1,TABLES,FANCY)
L 3: SELL(PLANT1,DINSETS,FUNCTIONAL)
L 4: SELL(PLANT1,DINSETS,FANCY)
S 5: SELL(PLANT2,CHAIRS,FUNCTIONAL)
E 6: SELL(PLANT2,CHAIRS,FANCY)
L 7: SELL(PLANT3,TABLES,FUNCTIONAL)
L 8: SELL(PLANT3,TABLES,FANCY)
S 9: SELL(PLANT3,CHAIRS,FUNCTIONAL)
E 10: SELL(PLANT3,CHAIRS,FANCY)
NETINCOME 1: NETINCOME

#### Dictionary of Equations
OBJT 1: OBJT
R 1: RESOUREQ(PLANT1,LABOR)
E 2: RESOUREQ(PLANT1,TOP)
S 3: RESOUREQ(PLANT2,SMLLATHE)
O 4: RESOUREQ(PLANT2,LRGLATHE)
U 5: RESOUREQ(PLANT2,CARVER)
R 6: RESOUREQ(PLANT2,LABOR)
E 7: RESOUREQ(PLANT2,TOP)
Q 8: RESOUREQ(PLANT3,SMLLATHE)
R 9: RESOUREQ(PLANT3,LRGLATHE)
E 10: RESOUREQ(PLANT3,CARVER)
S 11: RESOUREQ(PLANT3,LABOR)
O 12: RESOUREQ(PLANT3,TOP)
LINKTABLE 1: LINKTABLE(FUNCTIONAL)
LINKTABLE 2: LINKTABLE(FANCY)
LINKCHAIR 1: LINKCHAIR(FUNCTIONAL)
LINKCHAIR 2: LINKCHAIR(FANCY)
TRNCHAIREQ 1: TRNCHAIREQ(PLANT2,FUNCTIONAL)
TRNCHAIREQ 2: TRNCHAIREQ(PLANT2,FANCY)
TRNCHAIREQ 3: TRNCHAIREQ(PLANT3,FUNCTIONAL)
TRNCHAIREQ 4: TRNCHAIREQ(PLANT3,FANCY)
TRNTABLEEQ 1: TRNTABLEEQ(PLANT3,FUNCTIONAL)
TRNTABLEEQ 2: TRNTABLEEQ(PLANT3,FANCY)
```

**Table 19.1** PUT file Output for TRANSPRT example used in Chapter 6.

Report of Output from Solve of Transport Model  
Run Done on 01/18/94 at 10:01:13

Commodity Movements Between Cities  
all units in tons

Origin City	Destination City				Total
	MIAMI	HOUSTON	MINEPLIS	PORTLAND	
NEWYORK	30	35	15	0	80
CHICAGO	0	0	75	0	75
LOSANGLS	0	40	0	50	90
	----	----	----	----	----
Total	30	75	90	50	245

Cost of Commodity Movements Between Cities

Origin City	Destination City	Quantity Shipped tons	Cost/unit \$/ton	Cost Incurred \$
NEWYORK	MIAMI	30	20	600
NEWYORK	HOUSTON	35	40	1400
NEWYORK	MINEPLIS	15	35	525
CHICAGO	MINEPLIS	75	20	1500
LOSANGLS	HOUSTON	40	35	1400
LOSANGLS	PORTLAND	50	40	2000
				----
Total Cost of Shipping				7425

Report on Status of Commodity Usage by Plant

Plant	Quantity Available in tons	Quantity Shipped in tons	Value of More Supply in \$ / ton
NEWYORK	100	80	0.00
CHICAGO	75	75	15.00
LOSANGLS	90	90	5.00

Report on Status of Supply By Market

Market	Quantity Needed in tons	Quantity Received in tons	Cost of Meeting Demand in \$ / ton
MIAMI	30	30	20.00
HOUSTON	75	75	40.00
MINEPLIS	90	90	35.00
PORTLAND	50	50	45.00

Cost of Altering Commodity Movements Between Cities  
all units in \$/ton

Origin City	Destination City			
	MIAMI	HOUSTON	MINEPLIS	PORTLAND
NEWYORK	0.00	0.00	0.00	75.00
CHICAGO	45.00	35.00	0.00	40.00
LOSANGLS	75.00	0.00	40.00	0.00

**Table 19.2** Comparative Report Writer Code from EVPORTFO File

```
PARAMETER RISK AVER(S) RISK AVERSION COEFFICIENTS
/ R0 0.00000, R1 0.00025, R2 0.00050, R3 0.00075,
R4 0.00100, R5 0.00150, R6 0.00200, R7 0.00300,
R8 0.00500, R9 0.01000, R10 0.01100, R11 0.01250,
R12 0.01500, R13 0.02500, R14 0.05000, R15 0.10000,
R16 0.30000, R17 0.50000, R18 1.00000, R19 2.50000,
R20 5.00000, R21 10.0000, R22 15. , R23 20.
R24 40. , R25 80./
```

```
PARAMETER OUTPUT(*,RAPS) RESULTS FROM MODEL RUNS WITH VARYING RAP
```

```
OPTION SOLPRINT = OFF;
LOOP (RAPS,RAP=RISK AVER(RAPS));
  SOLVE EVPORTFOL USING NLP MAXIMIZING OBJ ;
  VAR = SUM(STOCK, SUM(STOCKS,
INVEST.L(STOCK)* COVAR(STOCK,STOCKS) * INVEST.L(STOCKS))) ;
  OUTPUT("OBJ",RAPS)=OBJ.L;
  OUTPUT("RAP",RAPS)=RAP;
  OUTPUT(STOCKS,RAPS)=INVEST.L(STOCKS);
  OUTPUT("MEAN",RAPS)=SUM(STOCKS, MEAN(STOCKS) *
INVEST.L(STOCKS));
  OUTPUT("VAR",RAPS) = VAR;
  OUTPUT("STD",RAPS)=SQRT(VAR);
  OUTPUT("SHADPRICE",RAPS)=INVESTAV.M;
  OUTPUT("IDLE",RAPS)=FUNDS-INVESTAV.L
);
DISPLAY OUTPUT;
```



Appendix I: Using Summation Notation With GAMS .....	AI-1
AI.1 Summation Mechanics .....	AI-1
AI.1.1 Sum of an Item. ....	AI-1
AI.1.2 Multiple Sums .....	AI-2
AI.1.3 Sum of Two Items .....	AI-2
AI.2 Summation Notation Rules .....	AI-3
AI.2.1 For a Scalar Equation .....	AI-3
AI.2.2. For a Family of Equations .....	AI-4
AI.4 Defining and Using Variables .....	AI-7
AI.5 Equations .....	AI-8
AI.6 Cautions and Extensions .....	AI-9

## Appendix I: Using Summation Notation With GAMS

Summation notation is difficult for some students to use and follow. Here we present notes on the mechanics of summation notation usage and some rules for proper use. This discussion is cast within the GAMS framework with presentation equivalents of common summation expressions and error messages caused by improper summation. All of the GAMS statements used herein are shown in Table 1 and are in file NOTATION.

### AI.1 Summation Mechanics

Summation notation is a short hand way of expressing sums of algebraic terms involving subscripted items. In order to cover the mechanics of summation notation it is useful to have a set of subscripted items and associated numerical values. Thus, let us define some data

$$\begin{array}{l} x_1 = 1 \quad y_{11} = 2 \quad y_{12} = 3 \\ x_2 = 2 \quad y_{21} = 4 \quad y_{22} = 1 \\ x_3 = 3 \quad y_{31} = 1 \quad y_{32} = 4. \end{array}$$

Now let us define a variety of summation expressions.

#### AI.1.1 Sum of an Item.

Suppose we wished to sum all values of x. This would be written as

$$\sum_{i=1}^3 x_i = x_1 + x_2 + x_3 = 1 + 2 + 3 = 6$$

or in GAMS

$$\text{SUM1} = \text{SUM}(I, X(I));$$

For short hand purposes if i was to be summed over all possible values, we would write this as

$$\sum_i x_i.$$

We might also express a sum as follows which indicates all of the i are summed over except i=3

$$\sum_{i \neq 3} x_i.$$

In GAMS, this is more difficult to express where one has to write a conditional (\$) operation or define a subset as follows

$$\text{SUM1} = \text{SUM}(I\$(ORD(I.NE.3)), X(I));$$

or

$$\begin{array}{l} \text{SET SUBSETI}(I) \quad /1, 2/; \\ \text{SUM1} = \text{SUM}(\text{SUBSETI}, X(\text{SUBSETI}(I))); \end{array}$$

#### AI.1.2 Multiple Sums

Sums over two indices consider all combinations of those items

$$\sum_{i,j} y_{ij} = y_{11} + y_{12} + y_{21} + y_{22} + y_{31} + y_{32} = 15.$$

The equivalent GAMS expression is

$$\text{SUM2} = \text{SUM}((\text{I},\text{J}), \text{Y}(\text{I},\text{J}));$$

### AI.1.3 Sum of Two Items

Suppose we wished to sum over two items completely where they shared a subscript

$$\sum_i (x_i + \sum_j y_{ij}) = \sum_i x_i + \sum_j y_{1j} + \sum_j y_{2j} + \sum_j y_{3j} = 21.$$

The equivalent GAMS expression is as follows

$$\text{SUM3} = \text{SUM}(\text{I}, \text{X}(\text{I}) + \text{SUM}(\text{J}, \text{Y}(\text{I}, \text{J})));$$

or

$$\text{SUM3} = \text{SUM}(\text{I}, \text{X}(\text{I})) + \text{SUM}((\text{I},\text{J}), \text{Y}(\text{I},\text{J}));$$

On the other hand, if we wished to sum the results only for the  $i^{\text{th}}$  element and call it  $A_i$  then

$$A_i = x_i + \sum_j y_{ij} = x_i + y_{i1} + y_{i2}$$

or in GAMS

$$A(\text{I}) = \text{X}(\text{I}) + \text{SUM}(\text{J}, \text{Y}(\text{I},\text{J}));$$

which would yield a vector [ 6 , 7 , 8 ] of results.

Sums over common subscripts can be collapsed or taken apart

or

$$\sum_i (x_i + z_i) = \sum_i x_i + \sum_i z_i$$

$$\text{SUM4} = \text{SUM}(\text{I}, \text{X}(\text{I}) + \text{Z}(\text{I}));$$

or

$$\text{SUM4} = \text{SUM}(\text{I}, \text{X}(\text{I})) + \text{SUM}(\text{I}, \text{Z}(\text{I}));$$

## AI.2 Summation Notation Rules

Certain rules apply when writing summation notation equations. The applicable rules depend on whether the final result is an unsubscripted scalar or a subscripted family of results determined by multiple equations.

### AI.2.1 For a Scalar Equation

All subscripts must be dealt with in each term. Thus, it is proper to define the equation

$$B = \sum_{i,j,k} p_{ijk} + \sum_{m,n} q_{mn}$$

However, the following equations are wrong

$$B2 = \sum_{i,j,k} p_{ijk} \sum_{m,n} q_{mn}$$

$$B3 = \sum_{j,i} p_{ijk} \sum_{m,n} q_{mn}$$

In the case of the first equation, the result would really have the subscripts i,j,k,m,n, while the second equation result would have to have a k subscript on B3 or a sum over k to be proper.

Equivalent GAMS commands for the above equation expressions are

```
EQB1.. B1 =E= SUM((I,J,K),P(I,J,K)) + SUM((M,N), Q(M,N));
EQB2.. B2 =E= P(I,J,K) + Q(M,N);
EQB3.. B3 =E= SUM((I,J), P(I,J,K)) + SUM((M,N), Q(M,N));
```

Here, the first equation expression is correct, while the last two equation expressions are incorrect. If you run GAMS with the above commands, you would encounter GAMS error messages \$149 which says "UNCONTROLLED SET ENTERED AS CONSTANT" meaning that you have not somehow dealt with all the subscripts in the equation.

### AI.2.2. For a Family of Equations

Several rules apply when one is working with a family of equations.

1. The members of the family must be specified with an indication of the subscripts which define each equation. This is done by indicating all the conditions for which the equations exist in a "for" condition. For example, suppose we define an equation which sets all C's equal to 2. This is done by saying

$$C_i = 2 \text{ for all } i \text{ or } C_i = 2 \text{ for } i = 1, 2, \dots n.$$

Similarly, if we wish to set a 2 dimensional variable equal to a constant, we would state

$$D_{ij} = 2 \text{ for all } i \text{ and } j,$$

while stating that for each row of the matrix  $E_{ij}$  we have the same values  $F_i$  is defined by

$$E1_{ij} = F_i \text{ for all } i \text{ and } j.$$

The equivalent GAMS commands for the above expressions are

```
EQUATIONS
      EQC(I)          EQUATION C
      EQD(I,J)       EQUATION D
      EQE1(I,J)      EQUATION E1;
EQC(I).. C(I) =E= 2;
EQD(I,J).. D(I,J) =E= 2;
EQE1(I,J).. E1(I,J) =E= F(I);
```

On the other hand, it is wrong to state

$$E2_{ij} = 2$$

without conditions on i and j. The equivalent GAMS commands for the above incorrect expressions are

```
EQUATION
      EQE2          EQUATION E2;
```

$$\text{EQE2..} \quad \text{E2(I,J) =E= 2;}$$

Here you would get error message \$149 which says "UNCONTROLLED SET ENTERED AS CONSTANT."

- When writing an equation with a for statement, all subscripts which are not in the for statement must be summed over. Consequently, it is proper to write

$$\sum_{j,k} p_{ijk} = G1_i \quad \text{for all } i$$

$$\sum_k p_{ijk} = H1_i \quad \text{for all } i \text{ and } j$$

but improper to write

$$p_{ijk} = G2_i \quad \text{for all } i$$

$$\sum_k p_{ijk} = H2_i \quad \text{for all } i.$$

The equivalent GAMS commands for the above equations are

```

EQUATIONS
    EQG1(I)      EQUATION G1
    EQH1(I,J)    EQUATION H1
    EQG2(I)      EQUATION G2
    EQH2(I)      EQUATION H2;
EQG1(I)..      G1(I) =E= SUM((J,K), P(I,J,K));
EQH1(I,J)..    H1(I,J) =E= SUM(K, P(I,J,K));
EQG2(I)..      G2(I) =E= P(I,J,K);
EQH2(I)..      H2(I) =E= SUM(K, P(I,J,K));

```

in which the first two equations are correct, while the last two equations are wrong and error messages \$149 "UNCONTROLLED SET ENTERED AS CONSTANT" would again be realized.

- In any term of an equation, the result after executing the mathematical operations in that term must be of a dimension less than or equal to the family definition in the for statement. For example, it is proper to write

$$\sum_{j,k} p_{ijk} = L1 \quad \text{for all } i$$

$$\sum_{j,k} r_{ijkm} \% \sum_j s_{ijm} = N_{im} \quad \text{for all } i \text{ and } m$$

but wrong to write

$$p_{ijk} = L2 \quad \text{for all } i.$$

Thus, for the following expressions, the first two equations are appropriate but the last equation would give you error message \$149 "UNCONTROLLED SET ENTERED AS CONSTANT."

```

EQUATION
EQLI(I)... LI(I) =E= SUM((J,K), P(IJK));
EQN(I,M)... N(I,M) =E= SUM((J,K), R(I,J,K,M))
+ SUM (J,S(I,J,M));
EQL2(I)... L2 =E= P(I,J,K);

```

4. When the dimension is less than the family definition this implies the same term appears in multiple equations. For example, in the equation

$$2 + \sum_j p_{ijk} + \sum_j s_{ijm} = O_{im} \quad \text{for all } i \text{ and } m,$$

the 2 term appears in every equation and the sum involving p is common when m varies.

```

EQUATION
EQQ(I,M) EQUATION O;
EQQ(I,M).. 2 + SUM((J,K), P(I,J,K)) + SUM(J, S(I,J,M)) =E= O(I,M);

```

5. In an equation you can never sum over the parameter that determines the family of equations. It is certainly wrong to write

$$\sum_k p_{ijk} = W_i \quad \text{for all } i.$$

Or, equivalently, the following expression will result in error

```

EQW(I)... W(I) =E= SUM(I,J,K), P(I,J,K));

```

### AI.3 Defining Subscripts

In setting up a set of equations and variables use the following principles. Define a subscript for each physical phenomena set which has multiple members, i.e.,

Let  $i$  denote production processes of which there are I  
 $j$  denote locations of which there are J  
 $k$  denote products of which there are K  
 $m$  denote sales locations of which there are M.

Equivalent GAMS commands are

```

SET I /1*20/
J /1*30/
K /1*5/
M /CHICAGO, BOSTON/;

```

Define different subscripts when you are either considering subsets of the subscript set or different physical phenomena.

### AI.4 Defining and Using Variables

1. Define a unique symbol with a subscript for each manipulatable item.

For example:

$p_{ijk}$  = production using process  $i$  at location  $j$  while producing good  $k$ .

Or, equivalently,

```

PARAMETER P(I,J,K)

```

or

PARAMETER PRODUCTION(PROCESS, LOCATION, GOOD)

Here, for documentation purposes, the second expression is preferred.

- 2. Make sure that variable has the same subscript in each place it occurs.

Thus it is proper to write

$$\text{Max } \sum_{i,j,k} t_{ijk} \quad \text{for all } k$$

but wrong to write

$$\text{Max } \sum_{i,j} t_{ij} \quad \text{for all } k$$

$$t_{ijk} = 0.$$

The second model would cause error message \$148 indicating "DIMENSION DIFFERENT."

- 3. The authors feel it is a bad practice to define different items with the same symbol but varying subscripts. We think you should never use the same symbol for two different items as follows

$u_{ij}$  = amount of tires transported from i to j and

$u_{kj}$  = amount of chickens transported from k to j.

GAMS would not permit this, giving error \$150 "Symbolic Equations Redefined."

### AI.5 Equations

Modelers should carefully identify the conditions under which each equation exists and use subscripts to identify those conditions. We do not think modelers should try to overly compact the families of equations. For example, it is OK to define

$$\sum_j a_{ij} x_j \# b_i$$

for all i, where  $a_{ij}$  is use of water by period and labor by period, where i denotes water periods and labor periods and  $b_i$  simultaneously contains water and labor availability by period. But we find it is better to define

$$\sum_j d_{ij} x_j \# e_i$$

$$\sum_j f_{ij} x_j \# h_i$$

where i denotes period,

$d_{ij}$  denotes water use and  $e_i$  water availability,

$f_{ij}$  denotes labor use and  $h_i$  labor availability.

### AI.6 Cautions and Extensions

1. Be careful when you sum over terms which do not contain the subscript you are summing over. This is equivalent to multiplying a term by the number of items in the sum.

$$\sum_{i=1}^3 \sum_{j=1}^1 x_i = 3(2) = 6$$

Or, in GAMS

$$\text{SUM5A} = \text{SUM}(J, X("2"));$$

2. Be careful when you have a term in a family of equations which is of a lesser dimension than the family, this term will occur in each equation. For example, the expression

$$x_j = z_i \quad \text{for } i = 1, 2, 3$$

implies that simultaneously

$$\begin{aligned} x_j &= z_1 \\ x_j &= z_2 \\ x_j &= z_3. \end{aligned}$$

3. The same rules as outlined above apply to product cases

$$\sum_{i=1}^3 x_i = x_1 * x_2 * x_3.$$

Or, equivalently,

$$\text{PRODUCTX} = \text{PROD}(I, X(I));$$

4. The following relationships also hold for summation

- a.  $\sum_{i=1}^n K x_i = K \sum_{i=1}^n x_i$
- b.  $\sum_{i=1}^n K P = K \sum_{i=1}^n P = K n P$
- c.  $\sum_{i=1}^n (v_{ij} \% y_{ij}) = \sum_{i=1}^n v_{ij} \% \sum_{i=1}^n y_{ij}$
- d.  $\sum_{i=1}^n (x_i \% y_{ij}) = \sum_{i=1}^n x_i \% \sum_{i=1}^n y_{ij}$  when  $j = 1, 2, \dots, n$



**Table 1. Sample GAMS Commands for Summation Notation Expressions**

```

1 *****
2 ** THIS FILE CONTAINS GAMS EXAMPLES IN SUPPORT **
3 ** OF THE NOTES USING THE SUMMATION NOTATION **
4 *****
5
6 SETS
7     I /1*3/
8     J /1*2/
9     K /1*2/
10    M /1*2/
11    N /1*3/
12
13 PARAMETERS
14
15     X(I) /1 1,2 2,3 3/
16     Z(I) /1 2,2 4,3 6/
17
18 TABLE Y(I,J)
19
20         1      2
21     1      2      3
22     2      4      1
23     3      1      4;
24
25 TABLE V(I,J)
26
27         1      2
28     1      2      3
29     2      4      1
30     3      1      4;
31
32 TABLE P(I, J, K)
33
34         1.1  1.2  2.1  2.2
35     1      1      3      5      7
36     2      2      4      6      8
37     3      1      2      3      4 ;
38
39 TABLE Q(M, N)
40
41         1      2      3
42     1      1      5     10
43     2     10      5      1;
44
45 *****
46 ** AI.1.1 SUM OF AN ITEM **
47 *****
48
49 PARAMETER
50     SUM1      SUM OF AN ITEM;
51     SUM1      = SUM(I, X(I));
52     DISPLAY SUM1;
53
54 *****
55 ** AI.1.2 MULTIPLE SUMS **
56 *****
57
58 PARAMETER
59     SUM2      MULTIPLE SUMS;
60     SUM2      = SUM((I,J), Y(I,J));
61     DISPLAY SUM2;

```

**Table 1.** Sample GAMS Commands for Summation Notation Expressions (continued)

```
62
63 *****
64 ** AI.1.3 SUM OF TWO ITEMS **
65 *****
66
67 PARAMETERS
68   SUM3A      SUM OF TWO ITEMS-1
69   SUM3B      SUM OF TWO ITEMS-1
70   A(I)       SUM OF TWO ITEMS-2
71   SUM4A      SUM OF TWO ITEMS-3
72   SUM4B      SUM OF TWO ITEMS-3;
73   SUM3A      = SUM(I, X(I)+SUM(J, Y(I, J)));
74   SUM3B      = SUM(I, X(I)) + SUM ((I,J), Y(I,J));
75   A(I)       = X(I) + SUM(J, Y(I,J));
76   SUM4A      = SUM(I, X(I)+Z(I));
77   SUM4B      = SUM(I, X(I)) + SUM(I, Z(I));
78   DISPLAY SUM3A, SUM3B, A, SUM4A, SUM4B;
79
80 *****
81 ** AI.2.1 FOR A SCALAR EQUATION **
82 *****
83
84 PARAMETERS
85   B1      SUM FOR A SCALAR EQUATION-1;
86   B1 = SUM((I,J,K), P(I,J,K)) + SUM((M,N), Q(M,N));
87   DISPLAY B1;
88
89 * $ONTEXT
90 * THE FOLLOWING SUMMATION NOTATIONS ARE INCORRECT
91 * IF YOU TURN THESE COMMANDS ON, YOU WILL ENCOUNTER
92 * ERROR MESSAGES
93 * PARAMETERS
94 *   B2      SUM FOR A SCALAR EQUATION-2
95 *   B3      SUM FOR A SCALAR EQUATION-3;
96 *   B2 = P(I,J,K) + Q(M,N);
97 *   B3 = SUM((I,J), P(I,J,K)) + SUM((M,N), Q(M,N));
98 *   DISPLAY B2, B3;
99 * $OFFTEXT
100
101 *****
102 ** A.I.2.2 FOR A FAMILY OF EQUATIONS **
103 *****
104
105 VARIABLES      C(I), D(I,J), E1(I,J), F(J);
106 EQUATIONS
107               EQC(I)      EQUATION C
108               EQD(I,J)    EQUATION D
109               EQE1(I,J)   EQUATION E1;
110   EQC(I)..    C(I) =E= 2;
111   EQD(I,J).. D(I,J) =E= 2;
112   EQE1(I,J).. E1(I,J) =E= F(J);
113
114 * $ONTEXT
115 * THE FOLLOWING EXPRESSION IS INCORRECT
116 * ERROR MESSAGES WILL BE ENCOUNTERED
117 * VARIABLES E2(I,J);
```

**Table 1.** Sample GAMS Commands for Summation Notation Expressions (continued)

```
118 * EQUATION
119 *       EQE2      EQUATION E2;
120 *       EQE2..    E2(I,J) =E= 2;
121 * $OFFTEXT
122
```

```

123 VARIABLES G1(I), H1(I,J);
124 EQUATIONS
125     EQG1(I)    EQUATION G1
126     EQH1(I,J) EQUATION H1;
127     EQG1(I).. G1(I) =E= SUM((J,K), P(I,J,K));
128     EQH1(I,J).. H1(I,J) =E= SUM(K, P(I,J,K));
129
130 * $ONTEXT
131 * THE FOLLOWING EXPRESSIONS ARE INCORRECT
132 * ERROR MESSAGES WILL BE ENCOUNTERED
133 * VARIABLES G2(I), H2(I);
134 * EQUATIONS
135 *     EQG2(I)    EQUATION G2
136 *     EQH2(I)    EQUATION H2;
137 *     EQG2(I).. G2(I) =E= P(I,J,K);
138 *     EQH2(I).. H2(I) =E= SUM(K, P(I,J,K));
139 * $OFFTEXT
140
141 VARIABLES L1(I), U(I,M), R(I,J,K,M), S(I,J,M);
142 EQUATIONS
143     EQL1(I)    EQUATION L1
144     EQN(I,M)   EQUATION N;
145     EQL1(I).. L1(I) =E= SUM((J,K), P(I,J,K));
146     EQN(I,M).. U(I,M) =E= SUM((J,K),R(I,J,K,M)) + SUM(J, S(I,J,M));
147
148 * $ONTEXT
149 * THE FOLLOWING EXPRESSIONS ARE INCORRECT
150 * ERROR MESSAGES WILL BE ENCOUNTERED
151 * VARIABLES L2;
152 * EQUATIONS
153 *     EQL2(I)    EQUATION L2;
154 *     EQL2(I).. L2 =E= P(I,J,K);
155 * OFFTEXT
156
157 VARIABLE O(I,M);
158 EQUATION
159     EQO(I,M) EQUATION O;
160     EQO(I,M).. 2 + SUM((J,K), P(I,J,K)) + SUM(J, S(I,J,M)) =E=
O(I,M);
161
162
163 * $ONTEXT
164 * THE FOLLOWING EXPRESSION IS INCORRECT
165 * GAMS ERROR MESSAGES WILL BE ENCOUNTERED
166 * VARIABLE W(I);
167 * EQUATION
168 *     EQW(I)    EQUATION W;
169 *     EQW(I).. W(I) =E= SUM((I,J,K), P(I,J,K));
170 * $OFFTEXT
171

```

**Table 1.** Sample GAMS Commands for Summation Notation Expressions (continued)

---

```

172 *****
173 ** AI.4 DEFINING AND USING VARIABLES **
174 *****
175
176 VARIABLES
177     OBJ1    OBJECTIVE FUNCTION VALUE
178     T(I,J,K) DECISION VARIABLE;
179 EQUATIONS
180     OBJFUNC1 OBJECTIVE FUNCTION
181     CONST(K) CONSTRAINT;
182     OBJFUNC1.. OBJ1 =E= SUM((I,J,K), T(I,J,K));
183     CONST(K).. SUM((I,J), T(I,J,K)) =E= 3;

```

```

184 MODEL EXAMPLE1 /ALL/;
185 SOLVE EXAMPLE1 USING LP MAXIMIZING OBJ1;
186 DISPLAY T.L;
187
188 * $ONTEXT
189 * THE FOLLOWING COMMANDS ARE INCORRECT
190 * THEY WILL RESULT IN ERROR MESSAGES
191 * VARIABLES
192 *   OBJ2           OBJECTIVE FUNCTION VALUE
193 *   TT(I,J,K)     DECISION VARIABLE;
194 * POSITIVE VARIABLE TT;
195 * EQUATIONS
196 *   OBJFUNC2      OBJECTIVE FUNCTION
197 *   CONSTT(K)     CONSTRAINT;
198 *   OBJFUNC2..   OBJ2 =E= SUM((I,J), TT(I,J));
199 *   CONSTT(K)..  SUM((I,J), TT(I,J,K)) =E= 3;
200 * MODEL EXAMPLE2 /ALL/;
201 * SOLVE EXAMPLE2 USING LP MAXIMIZING OBJ2;
202 * DISPLAY TT.L;
203 * $OFFTEXT
204
205 *****
206 ** AI.6 CAUTIONS AND EXTENSIONS **
207 *****
208
209 PARAMETER
210   SUM5A CAUTIONS AND EXTENSIONS-1;
211   SUM5A = SUM(J, X("2"));
212   DISPLAY SUM5A;
213
214 PARAMETER
215   PRODUCT6 CAUTIONS AND EXTENSIONS-2;
216   PRODUCT6 = PROD(I, X(I));
217   DISPLAY PRODUCT6;
218
219 PARAMETERS
220   SUM7A CAUTIONS AND EXTENSIONS-3
221   SUM7B CAUTIONS AND EXTENSIONS-3
222   SUM8A CAUTIONS AND EXTENSIONS-4
223   SUM8B CAUTIONS AND EXTENSIONS-4
224   SUM8C CAUTIONS AND EXTENSIONS-4
225   SUM9A CAUTIONS AND EXTENSIONS-5
226   SUM9B CAUTIONS AND EXTENSIONS-5

```

**Table 1.** Sample GAMS Commands for Summation Notation Expressions (continued)

---

```

227   SUM10A CAUTIONS AND EXTENSIONS-6
228   SUM10B CAUTIONS AND EXTENSIONS-6;
229   SUM7A = SUM(I, 5*X(I));
230   SUM7B = 5*SUM(I, X(I));
231   SUM8A = SUM(I, 5*10);
232   SUM8B = 5*SUM(I, 10);
233   SUM8C = 5*3*10;
234   SUM9A = SUM((I,J), V(I,J)+Y(I,J));
235   SUM9B = SUM((I,J), V(I,J)) + SUM((I,J), Y(I,J));
236   SUM10A = SUM((I,J), X(I)+Y(I,J));
237   SUM10B = 2*SUM(I, X(I)) + SUM((I,J), Y(I,J));
238   DISPLAY SUM7A, SUM7B, SUM8A, SUM8B, SUM8C,
239           SUM9A, SUM9B, SUM10A, SUM10B;

```



## APPENDIX II: EXECUTING GAMS

GAMS versions exist for many different computer platforms. These authors have had experience with micro computer versions for PC's, workstation versions for Sun and HP systems and mainframe versions for IBM and CRAY machines. The basic strategy is almost the same for all of these machines. Thus a simple overview discussion will be presented with particulars only drawn out for PC and UNIX based machines.<sup>23</sup>

GAMS operates in a batch mode. Thus, an ASCII file describing the problem must be created with a text editor, word processor or other means. In turn, assuming the GAMS input file is called "filename", the file is executed using the command

```
GAMS filename
```

In turn, program output appears on the file filename.lst.<sup>24</sup>

GAMS also allows the user control over a number of options involved with running the program. One may control page width, page length, saves and restarts among many other things. The files GAMSPARM.TXT and GAMSPARM.DOC details many of the other options. The general PC version of the call is

```
COMMAND \C GAMS FILENAM PW=90 PS=80 S=NEWSAV R=OLDSAV O=DIVERT
```

where the items are:

COMMAND \C	a line needed to make restarts work properly with DOS
FILENAM	the name of the file of GAMS input instructions to be executed
PS=number	specifies the page length be used in the output file
PW=number	specifies the page width be used in the output file
S=name	file name used for saved work files (no file name extension is allowed)
R=name	file name used for work files the program resumes from (no file extension is allowed)
O=filename	name of file used for diverted output.

The equivalent UNIX command is

```
gams filename -pw 90 -ps 80 -s newsav -r oldsav -o divert
```

---

<sup>23</sup> Users of different systems should consult the README file distributed with GAMS as well as any release notes and Appendix A of the GAMS manual.

<sup>24</sup> There are other variants of these naming conventions. Users on non-UNIX or PC machines should consult Appendix A of Brooke, Kendrick, and Meeraus or the GAMS installation guide for the system to be used.

