72-11,591

KONNO, Hiroshi, 1940-
BILINEAR PROGRAMMING.

Stanford University, Ph.D., 1971
Mathematics

University Microfilms, A XEROX Company, Ann Arbor, Michigan

THIS DISSERTATION HAS BEEN MICROFILMED EXACTLY AS RECEIVED

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

# BILINEAR PROGRAMMING

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF OPERATIONS RESEARCH

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

By

Hiroshi Konno

August 1971

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

George B. Dantzig

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Richard W Cottle

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Approved for the University Committee on the Graduate Division:

Dean of Graduate Studies



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

PLEASE NOTE:

Some pages have indistinct print. Filmed as received.

UNIVERSITY MICROFILMS.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

ACKNOWLEDGEMENTS

It is a great pleasure to express my deepest indebtedness to my principal adviser Professor George B. Dantzig for his continuous guidance and encouragement through the entire period of this research, without which this work would not have been completed. His influence goes far beyond this dissertation and has affected my basic way of thinking.

I would also like to thank Professor Richard W. Cottle for his encouragement and valuable consultation from the very early stage of this research.

The extent to which this dissertation is correct in content and style, I owe to the members of my reading committee, Professors George B. Dantzig, Richard W. Cottle, and B. Curtis Eaves, and to my office mate, Mr. Marion Reynolds, Jr.

My special thanks goes to Professor Sigeiti Moriguti who was my former adviser at the University of Tokyo and who introduced me to Stanford University.

To Mrs. Iona Williams, who typed this dissertation with tremendous skill and effort, I want to express deep gratitude.

Also I wish to thank the following institutions and agencies which enabled me to complete three years of my Ph.D. work at Stanford: Central Research Institute of Electric Power Industries (JAPAN), National Science Foundation, Atomic Energy Commission, and the Office of Naval Research.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

TABLE OF CONTENTS


| | | Page |
|---|---|---|
| | Acknowledgements | iii |
| | Introduction and Summary | 1 |
| PART I. | Algorithm for Solving Bilinear Programs | 6 |
| Chapter | | |
| 1 | EXTENDED LINEAR PROGRAMS AND THE BILINEAR PROGRAMMING PROBLEMS | 7 |
| 2 | PRELIMINARIES AND INITIAL LOCALLY MAXIMUM VERTEX | 14 |
| | 2.1 Preliminary Properties of the Bilinear Programming Problem | 14 |
| | 2.2 Computation of the Initial Locally Maximum Vertex | 22 |
| 3 | CUTTING PLANES AT A LOCALLY MAXIMUM VERTEX | 27 |
| | 3.1 Introduction | 27 |
| | 3.2 The cuts $g^t x + h^t y \geq \tau$ | 29 |
| | 3.3 The cuts $g^t x \geq \tau_x$ , $h^t y \geq \tau_y$ | 39 |
| 4 | SUCCESSIVE LOCALLY MAXIMUM VERTICES AND THE FINITE CONVERGENCE TO AN EXTREME $\epsilon$-OPTIMAL SOLUTION | 49 |
| | 4.1 Computation of Successive Locally Maximum Vertices | 49 |
| | 4.2 Finite Convergence of the Algorithm to an Extreme $\epsilon$-Optimal Solution | 56 |
| 5 | EXTENSIONS OF BILINEAR PROGRAMMING | 62 |
| | 5.1 Extended Bilinear Programming Problems | 62 |
| | 5.2 Other Extensions | 74 |
| PART II. | Application of Bilinear Programming | 78 |
| | Introduction | 79 |
| Chapter | | |
| 1 | GAME THEORETIC APPLICATIONS | 82 |




Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

| Chapter | | Page |
|---|---|---|
| | 1.1 Equilibrium Point of a Constrained Bimatrix Game | 82 |
| | 1.2 Two-Move Game with Perfect Information | 89 |
| 2 | FINITE HORIZON MARKOVIAN DECISION PROBLEMS | 92 |
| | 2.1 Multi-Stage Markovian Assignment Problem | 92 |
| | 2.2 Multi-Stage Production and Sales Optimization of a Monopolistic Enterprise | 98 |
| 3 | COMPLEMENTARY (ORTHOGONAL) PLANNING PROBLEMS | 102 |
| | 3.1 Introduction | 102 |
| | 3.2 Complementary Flows in a Network | 104 |
| | 3.3 Orthogonal Production Scheduling | 114 |
| 4 | REDUCTION OF A CERTAIN CLASS OF MATHEMATICAL PROGRAMMING PROBLEMS TO BILINEAR PROGRAMMING PROBLEMS | 117 |
| | 4.1 Maximization of a Convex Quadratic Function over a Polyhedral Convex Set | 117 |
| | 4.2 0-1 Integer Programming Problem | 128 |
| 5 | MISCELLANEOUS BILINEAR PROGRAMMING PROBLEMS | 133 |
| | 5.1 Reduction of a Sparse Matrix into an Almost-Triangular Matrix | 133 |
| | 5.2 A Location Problem on a Rectangular Network | 140 |
| Appendix A. | A NUMERICAL EXAMPLE | 145 |
| Appendix B. | THE TRANSFORMATION OF A GENERAL BLP INTO THE STANDARD FORM | 150 |
| References | | 152 |




Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Notation

All the vectors in this paper are assumed to be columnar unless otherwise indicated and the superscript $t$ refers to the transpose of a vector or a matrix.

$0_m$ and $e_m$ will be used as the $m$ dimensional vectors of zeroes and ones, respectively, while $0$ will be used as the vector or the matrix of zeroes of appropriate dimension when there is no possible confusion.

Two m-dimensional vectors $x = (x_1,\ldots,x_m)^t$ and $y = (y_1,\ldots,y_m)^t$ satisfy the relation $x \geq y (x > y)$ if, and only if $x_i \geq y_i (x_i > y_i)$ for $i=1,\ldots,m$.

Following abbreviations will be used frequently:

LP ----- 'linear programming (problem)'

QP ----- 'quadratic programming (problem)'

LCP ---- 'linear complementarity (problem)'

BLP ---- 'bilinear programming (problem)'

EBLP --- 'extended bilinear programming (problem)'

ext C -- 'the set of extreme points of the convex set C'

Bracket [ ] is used for bibliographical reference.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

# INTRODUCTION AND SUMMARY

## Introduction

This paper deals with an algorithm for and the applications of the bilinear programming problem:

$$\begin{cases} \underset{x,y}{\text{maximize}} & c^t x + d^t y + x^t C y \\ \text{subject to} & Ex \leq e \ , \quad x \geq 0 \\ & Fy \leq f \ , \quad y \geq 0 \end{cases}$$

where $x$ and $y$ are $m$ and $n$ dimensional vectors, respectively, and $c$ , $d$ , $e$ , $f$ , $C$ , $E$ and $F$ are known vectors or matrices of appropriate dimensions. This is a special case of the general (nonconcave) quadratic programming problem:

$$\begin{cases} \underset{z}{\text{maximize}} & q^t z + \frac{1}{2} z^t Q z \\ \text{subject to} & Az \leq b \ , \quad z \geq 0 \end{cases}$$

The concave quadratic programming problem, i.e., the maximization of a concave quadratic function subject to linear constraints, has been subject to intensive research in mathematical programming ever since the linear programming problem was successfully solved by the simplex algorithm. It has been fully solved both theoretically and algorithmically via Kuhn-Tucker necessary and sufficient condition [K-2] . There now exist several efficient (finite) simplex-like algorithms for this class of problems.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

On the other hand, if the quadratic objective function is not (pseudo-) concave, then it can have multiple local maxima. Determining a global maximum becomes inherently more difficult and not much results have been obtained in this area. A limited number of references for this problem are available which may be roughly classified as follows:

(i) Construction of a general algorithm which can solve any quadratic programming problem.
   (a) Cutting plane approach ([R-1], [R-2]).
   (b) Enumeration of all Kuhn-Tucker (stationary) points ([M-4]).

(ii) Identification of the class of quadratic functions which guarantees a unique local maximum (in the feasible region) ([C-6], [C-7] etc.).

(iii) Development of special algorithms for special problems ([G-2], [T-3]).

The complete resolution of the general (non-concave) quadratic programming problem is important not only because it is the first step toward more general nonconcave maximization problems but also because a number of integer programming problems can be so formulated. If we look more closely at the integer problems, we find that many of them have some special structure in either the objective function or the constraint set (convex objective function, block angular constraint set, etc.) and it appears worthwhile to exploit this structure in developing special algorithm.

The development of special algorithms for solving efficiently structured mathematical programs has been pursued from the time when the



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

simplex algorithm was invented. The algorithm for the network-type problems, the decomposition technique for the block-angular structure and a number of partitioning procedures (see [D-3], [G-1], [L-1]) are examples. Judging from current research effort, it appears that research on special structures will become more and more important in the future both for linear and nonlinear programming provided the structures under consideration are general enough to have a wide range of applications.

This paper is an example of research in this direction. Although the structure of the bilinear programming problem may appear at first glance to be too specialized to allow many applications, it turns out to be a very fruitful model for some class of real world problems. In fact the bilinear program can be interpreted as the closest extension of the linear programming problem.

Historically, this problem was first treated explicitly by H. Mills [M-3] in connection with finding an equilibrium point of a bimatrix game. In 1964, O. Mangasarian and H. Stone [M-2] and O. Mangasarian [M-1] treated the same problem and proposed the applications of (i) the gradient projection method or (ii) the complete enumeration technique. However, (i) does not guarantee the convergence to the global optimal solution for the general bilinear programs and (ii) is not very efficient. Later, the general bilinear programming problem was treated by M. Altman [A-1] and by A. V. Cabot and R. L. Francis [C-1], but all of whom addressed themselves to the problem of finding an algorithm for obtaining a local but not necessarily a global optimum.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Summary

In Part I of this paper, we develop an algorithm for obtaining a global optimum of the bilinear programming problem. Basically, it is the combination of the powerful simplex algorithm and the extension and elaboration of the cutting plane algorithm proposed by K. Ritter ([R-1], [R-2]). It must be emphasized that our algorithm uses nothing but the simplex algorithm and that no elaborate subproblem needs be solved even for constructing a cutting plane, so that it is expected to be a practical algorithm for this class of problems. It will be shown that this algorithm generates an $\epsilon$-optimal solution (and sometimes an optimal solution) in finitely many steps provided the feasible region is bounded. (The proof of finiteness for the case where the feasible region is unbounded remains as an open question.) In the final chapter, the extensions of the bilinear programming algorithm to more general problems are briefly discussed.

In Part II, we formulate a number of real world problems as bilinear programming problems and look into the application of the algorithm developed in Part I. Problems to be discussed are:

1. Game-Theoretic Problems
   (a) Equilibrium point of a constrained bimatrix game.
   (b) Two-move game with perfect information.
2. Finite-horizon Markovian Decision Problems
   (a) Multi-stage Markovian assignment problem.
   (b) Multi-stage production and sales optimization of a monopolistic enterprise.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

3. Complementary (Orthogonal) Planning Problems

   (a) Complementary flows in a network.

   (b) Orthogonal production scheduling.

4. Reduction of a Certain Class of Mathematical Programming Problems to Bilinear Programming Problems

   (a) Maximization of a convex quadratic function over a polyhedral convex set.

   (b) 0-1 integer programming problem.

5. Miscellaneous Bilinear Programming Problems

   (a) Reduction of a sparse matrix into an almost-triangular matrix by row and column permutations.

   (b) A location problem on a rectangular network.

This class of models seem to have very wide range of applications in the real world situations but none appear to have been treated in the literature or solved except by the complete enumeration technique.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

PART I

## Algorithm for Solving Bilinear Programs



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

## 1. EXTENDED LINEAR PROGRAMS AND THE BILINEAR PROGRAMMING PROBLEMS

In this chapter, we will define the bilinear programming problem as a natural extension of the linear programming problem and discuss its relationship to other programming problems.

The linear programming problem in its inequality form is defined by

$$\text{(1.1)} \qquad \begin{cases} \text{maximize} & p^t x \\ \text{subject to} & Ax \leq b \\ & x \geq 0 \end{cases},$$

where $x \in R^n$ is a variable vector and $b \in R^m$, $p \in R^n$ and $A \in R^{m \times n}$ are given vectors and a given matrix, respectively. The simplex algorithm due to G . Dantzig solves this class of problems very efficiently and there have been a good many extensions of this algorithm to more difficult problems such as integer programming problems, generalized linear programming problems, convex quadratic programming problems (see [D-1] for these extensions), and linear complementarity problems [C-4], etc.

Here, we will consider another type extension of the linear programming problem (1.1) to the case where $p$ is not necessarily fixed but can be chosen from a certain polyhedral convex set $P = \{p \in R^n \mid Lp \leq \ell, p \geq 0\}$, where $L \in R^{k \times n}$, $\ell \in R^k$, say. Then we have the following problem:



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$(1.2)\qquad \begin{cases} \underset{p,x}{\text{maximize}} & p^t x \\ \text{subject to} & Ax \le b \ , \quad x \ge 0 \\ & Lp \le \ell \ , \quad p \ge 0 \end{cases}$$

which we call an 'extended linear programming problem'.

Remark. Even if we vary $b$ in the polyhedral convex set together with $p$ in (1.1) the problem is still reducible to (1.2). It is also easy to see that if we only vary $b$ in a convex (not necessarily polyhedral) set instead of varying $p$, then the problem becomes a generalized linear program of P. Wolfe [D-1].

Problem (1.2) is sufficiently difficult and important to merit study for its own sake. Nevertheless, we will consider it as a special case of the 'bilinear programming problem' to be defined below:

Definition 1.1. The 'bilinear programming problem' in its standard form is a mathematical programming problem having the following structure

$$(1.3)\qquad \begin{cases} \underset{x,y}{\text{maximize}} & \varphi(x,y) = c^t x + d^t y + x^t C y \\ \text{subject to} & Ex \le e \ , \quad x \ge 0 \\ & Fy \le f \ , \quad y \ge 0 \end{cases}$$

where $x \in R^m$, $y \in R^n$ are variable vectors and $c \in R^m$, $d \in R^n$, $C \in R^{m \times n}$, $E \in R^{k_1 \times m}$, $F \in R^{k_2 \times n}$, $e \in R^{k_1}$, $f \in R^{k_2}$ are given vectors or matrices.

Obviously (1.2) is a psecial case of (1.3) where $m = n$, $c = 0$, $d = 0$ and $C = I$. For the convenience of the following presentation,



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

we will define the following sets

$$X_o = \{x \in R^m \mid Ex \leq e,\ x \geq 0\}$$
$$Y_o = \{y \in R^n \mid Fy \leq f,\ y \geq 0\} \tag{1.4}$$

When the constraints are stated as a system of linear inequalities and non-negative variables, we will say that the bilinear programming problem is in 'standard form'. We will deal exclusively with the bilinear programming problem in standard form (1.3) since with no loss of generality, there exist well known methods to convert a mixed system of equations and inequalities to standard form (see Appendix B).

Obviously the bilinear programming problem belongs to the class of general quadratic programming problems:

$$\begin{cases} \text{maximize} & f(z) = q^t z + \frac{1}{2} z^t Q z \\ \text{subject to} & Az \leq b \\ & z \geq 0 \end{cases} \tag{1.5}$$

where $z^t = (x^t, y^t)$, $q^t = (c^t, d^t)$ and

$$Q = \begin{pmatrix} 0 & C \\ C^t & 0 \end{pmatrix}, \quad A = \begin{pmatrix} E & 0 \\ 0 & F \end{pmatrix}, \quad b = \begin{pmatrix} e \\ f \end{pmatrix} \tag{1.6}$$

It is well known that if $Q$ is negative semi-definite, then $f(z)$ is a concave function of $z$, so that the associated Kuhn-Tucker point [K-2] as defined below gives a global optimal solution. Further, there exist several finite algorithms to find such a point ([C-4], [C-5], [W-4]).



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Definition 1.2. A pair of non-negative vectors $(z,w)$ satisfying the following (Kuhn-Tucker) conditions is called a Kuhn-Tucker point for (1.5) [K-2]:

$$
\begin{aligned}
& q + Qz - A^t w \leq 0 \quad , \quad Az - b \leq 0 \\
& z^t(q + Qz - A^t w) = 0 \quad , \quad w^t(Az - b) = 0
\end{aligned}
\tag{1.7}
$$

Unfortunately, however, it can be shown that $Q$ in (1.6) has $r$ positive eigenvalues and $r$ negative eigenvalues where $r$ is the rank of $C$, so that our objective function is neither concave nor convex for $C \neq 0$. In this case the Kuhn-Tucker conditions give only a stationary point and the algorithms quoted above will fail to find an optimal solution, in general.

In contrast to the concave case, research on the general quadratic programming problem is still in its preliminary stages with only a few published papers. As far as the author knows, there exists no algorithm which can solve a general quadratic programming problem in finitely many steps except the total enumeration of Kuhn-Tucker points. Other than this method, the cutting plane algorithm proposed by K. Ritter ([R-1], [R-2], [C-8]), seems to be the most important contribution. (The paper by R. Cottle and C. Mylander [C-8] is recommended for a better understanding of Ritter's original approach [R-1], [R-2].) Contrary to Ritter's original assertion (Lemma 5 [R-2]), his algorithm is not finite in general as shown by an example due to Zwart [Z-1]. It may still be possible to show, nevertheless,

(i) finite convergence to an $\epsilon$-optimal solution by some minor modification of the algorithm, and



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

(ii) the amount of computation need not be excessive.

This is exactly what we will show in the following chapters for the special case of a bilinear programming problem. To avoid confusion, we will give the definition of an $\epsilon$-optimal solution and of an optimal solution.

Definition 1.3. $(x^*, y^*)$ is an $\epsilon$-optimal solution of the bilinear programming problem (1.3) if $x^* \in X_o$, $y^* \in Y_o$ and $\varphi(x^*, y^*) \geq \varphi(x, y) - \epsilon$ for all $x \in X_o$ and all $y \in Y_o$. $(x^*, y^*)$ is said to be optimal if it is 0-optimal.

For most of the practical situations, we will be satisfied if we can get a feasible solution close enough to an optimal solution (relative to objective functional value) and the finite convergence to an $\epsilon$-optimal solution for any $\epsilon > 0$ is good enough from the practical and computational point of view.

The bilinear programming algorithm to be developed in the chapters below is a combination of the well established simplex algorithm and an elaboration of Ritter's cutting plane algorithm. It will generate an extreme $\epsilon$-optimal solution in finitely many steps for any $\epsilon > 0$ when $X_o$ and $Y_o$ are bounded. It must be stressed that the proposed algorithm uses nothing but the simplex algorithm (even for the construction of a cutting plane), so that it is expected to be practical. Moreover, it provides a useful upper bound for the objective function at each step as a byproduct. This upper bound is expected to enhance the practicality of our algorithm.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

In passing, we will consider the partial dual of the bilinear programming problem (1.3). By the duality theorem of linear programming, we have

$$\begin{aligned}
&\underset{x,y}{\text{maximize}} \{\varphi(x,y) \mid x \in X_o,\ y \in Y_o\} \\
&\quad = \max_x \left[ c^t x + \max_y \{(d + C^t x)^t y \mid Fy \le f,\ y \ge 0\} \;\middle|\; x \in X_o \right] \\
&\quad = \max_x \left[ c^t x + \min_z \{f^t z \mid F^t z \ge d + C^t x,\ z \ge 0\} \;\middle|\; x \in X_o \right]
\end{aligned}$$

if $Y_o$ is non-empty and bounded. Based upon this observation we will define the 'dual' of the BLP (1.3) as follows:

$$(1.8) \qquad \begin{cases} \max\limits_x \min\limits_z \ \zeta(x,z) = c^t x + f^t z \\ \text{subject to} \quad Ex \le e \\ \qquad\qquad\quad C^t x - F^t z \le -d \\ \qquad\qquad\quad x \ge 0,\ z \ge 0 \end{cases}$$

Theorem 1.1. Let $X_o$ and $Y_o$ be bounded. If $(x^*, y^*)$ is an $\epsilon$-optimal solution of the primal (1.3), then $(x^*, \bar{z}(x^*))$ is an $\epsilon$-optimal solution of the dual (1.8) where $\epsilon \ge 0$ and $\bar{z}(x) = \arg\min \{f^t z \mid F^t z \ge d + C^t x,\ z \ge 0\}$.

Proof.

It suffices to prove that

$$c^t x^* + f^t \bar{z}(x^*) \ge c^t x + f^t \bar{z}(x) - \epsilon$$

holds for all $x \in X_o$. By definition,



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$c^t x^* + f^t \bar{z}(x^*) = c^t x^* + \min\{f^t z \mid z \in Z(x^*)\}$$

$$= c^t x^* + \max\{(d + C^t x^*)^t y \mid y \in Y_o\}$$

$$\geq c^t x^* + d^t y^* + (x^*)^t C y^*$$

The last inequality follows from the fact that $y^* \in Y_o$. Also, we have for any $x \in X$ that

$$c^t x + f^t \bar{z}(x) = c^t x + \min\{f^t z \mid z \in Z(x)\}$$

$$= c^t x + \max\{(d + C^t x)^t y \mid y \in Y_o\}$$

$$\leq c^t x^* + d^t y^* + (x^*)^t C y^* + \epsilon$$

Combining these two inequalities, we have

$$c^t x^* + f^t \bar{z}(x^*) \geq c^t x^* + d^t y^* + (x^*)^t C y^*$$

$$\geq c^t x + f^t \bar{z}(x) - \epsilon \ ,$$

the desired inequality. Q.E.D.

As we shall see in Part II, there exists a number of practical applications of the primal or of the dual bilinear programming problem; both are solvable by our algorithm.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

## 2. PRELIMINARIES AND INITIAL LOCALLY MAXIMUM VERTEX

### 2.1 Preliminary Properties of the Bilinear Programming Problem

As we have seen in the previous chapter, the bilinear programming problem belongs to the class of non-concave quadratic programming problems which, in general, is difficult to solve. Note, however, that this problem reduces to a linear programming problem when we temporarily fix $x$ and maximize with respect to $y$, and conversely. This observation together with the total separability of the constraint sets into $X_o$ and $Y_o$ enables us to easily prove the following theorem (both the theorem and proof are known).

Theorem 2.1.1. If the bilinear programming problem (1.3) has a finite optimal solution, then there exists an optimal solution $(x^*, y^*)$ such that $x^*$ and $y^*$ are extreme points of $X_o$ and $Y_o$, respectively.

Proof. Let $(\bar{x}, \bar{y})$ be an optimal solution such that $\bar{x} \notin \text{ext } X_o$ and/or $\bar{y} \notin \text{ext } Y_o$. First we will assume that $\bar{x} \notin \text{ext } X_o$, $\bar{y} \notin \text{ext } Y_o$ and prove the existence of an extreme point optimal solution by constructing it starting from $(\bar{x}, \bar{y})$. (The result for the other cases will be obvious from the proof below.)

Let us consider the following linear programming problem

$$\underset{y}{\text{maximize}} \{\varphi(\bar{x}, y) \mid y \in Y_o\}$$

Since the original problem (1.3) is assumed to have a finite optimal solution, this subprobelem also must have a finite optimal solution.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

By the fundamental result of linear programming, there exists a point $y^* \in \text{ext } Y_o$ such that $\varphi(\bar{x}, y^*) \geq \varphi(\bar{x}, y)$ for all $y \in Y_o$. In particular, $\varphi(\bar{x}, y^*) \geq \varphi(\bar{x}, \bar{y})$. Since $(\bar{x}, \bar{y})$ is an optimal solution of (1.3), we have $\varphi(\bar{x}, \bar{y}) \geq \varphi(\bar{x}, y^*)$, so $(\bar{x}, y^*)$ is also an optimal solution. Next let us consider the second subproblem:

$$\underset{x}{\text{maximize}} \{\varphi(x, y^*) | x \in X_o\}$$

By the same reasoning as before, there exists a point $x^* \in \text{ext } X_o$ such that $\varphi(x^*, y^*) = \varphi(\bar{x}, y^*)$. It follows that $(x^*, y^*)$ is another optimal solution of (1.3) where $x^* \in \text{ext } X_o$ and $y^* \in \text{ext } Y_o$. Q.E.D.

Starting with any feasible solution $(\bar{x}, \bar{y})$, iterating the application of the above will in a finite number of steps produce a locally optimum extreme point solution. This fact is sometimes very important in applications (cf. Part II, Chapters 2, 3, and 5).

Theorem 2.1.1 shows that an optimal solution, if it exists, is given by a certain combination of the extreme points (or equivalently a combination of the basic feasible solutions) of $X_o$ and $Y_o$. The number of extreme points in each set is finite and the problem could be "solved" in finitely many steps by any finite algorithm ([B-2], e.g.) which enumerates, evaluates and compares all the extreme point solutions in the polyhedral convex set. In particular, if $X_o$ or $Y_o$, say $X_o$, has relatively few extreme points, then first enumerating all the extreme points $x^i$, i=1,...,p and finding the best corresponding extreme points $y^i$, i=1,...,p of $Y_o$ by solving p linear programs

$$\underset{y}{\text{maximize}} \{\varphi(x^i, y) | y \in Y_o\}, \quad i=1,\ldots,p$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

may be the most efficient algorithm since $\max_i \varphi(x^i, y^i)$ gives an optimal solution.

But this algorithm is impractical if the number of extreme points $p$ is large. Although the actual number of extreme points of $X_o$ and $Y_o$ may be of order $mk_1$ and $nk_2$ (see [K-1]) respectively, which is not very large, the numbers of potential candidates for an extreme point (i.e., the number of bases) of the sets $X_o$ and $Y_o$ may be as large as $\binom{m+k_1}{k_1}$ and $\binom{n+k_2}{k_2}$, respectively. These are both astronomical numbers even for a moderate size of $m$, $n$, $k_1$ and $k_2$ and an inordinate amount of computation will be required to find all the pairs of extreme points among these huge number of candidates. So we have to devise a more efficient and computationally feasible algorithm.

We now introduce the notion of the 'canonical form' of the bilinear programming problem (1.3) relative to a pair of extreme points $x^o, y^o$ of $X_o$ and $Y_o$. Let us define the slack variables $u$ and $v$ in a standard manner,

$$\begin{aligned} u &= e - Ex \quad , \quad u \geq 0 \\ v &= f - Fy \quad , \quad v \geq 0 \end{aligned} \tag{2.1.1}$$

and let

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad u = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \quad y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \quad v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

be the partition of the vectors where $(x_1, u_2)$ and $(y_1, v_2)$ are basic at $x^o, y^o$. Rewrite (2.1.1) in the partitioned form



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$
\begin{aligned}
u_1 &= e_1 - E_{11}x_1 - E_{12}x_2 \\
u_2 &= e_2 - E_{21}x_1 - E_{22}x_2 \\
v_1 &= f_1 - F_{11}y_1 - F_{12}y_2 \\
v_2 &= f_2 - F_{21}y_1 - F_{22}y_2
\end{aligned}
\tag{2.1.2}
$$

and solve for the basic variables $x_1$ and $y_1$ with respect to nonbasic variables using the first and the third equations of (2.1.2), i.e.,

$$
\begin{aligned}
x_1 &= E_{11}^{-1}(e_1 - E_{12}x_2 - u_1) \\
y_1 &= F_{11}^{-1}(f_1 - F_{12}y_2 - v_1)
\end{aligned}
$$

This is always possible since the submatrices $E_{11}$ and $F_{11}$ are bases and hence nonsingular by definition. Substituting these into (1.3) we obtain a new system

$$
\left\{
\begin{array}{l}
\text{maximize } \tilde{\varphi}(u_1,x_2,v_1,y_2) = \varphi(x^o,y^o) + \bar{c}_1^t u_1 + \bar{c}_2^t x_2 \\
\qquad\qquad + \bar{d}_1 v_1 + \bar{d}_2^t y_2 + (u_1^t, x_2^t)\begin{pmatrix} \bar{C}_{11} & \bar{C}_{12} \\ \bar{C}_{21} & \bar{C}_{22} \end{pmatrix}\begin{pmatrix} v_1 \\ y_2 \end{pmatrix} \\
\text{subject to } \bar{E}_1 u_1 + \bar{E}_2 x_2 \le \bar{e} \\
\qquad\qquad \bar{F}_1 v_1 + \bar{F}_2 y_2 \le \bar{f} \\
\qquad\qquad u_1 \ge 0\,,\ x_2 \ge 0 \\
\qquad\qquad v_1 \ge 0\,,\ y_2 \ge 0
\end{array}
\right.
\tag{2.1.3}
$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

where

$$\bar{c}_1 = -(E_{11}^{-1})^t(c_1 + C_{11}F_{11}^{-1}f_1)$$

$$\bar{c}_2 = c_2 - (E_{11}^{-1}E_{12})^t c_1 + \{C_{21} - (E_{11}^{-1}E_{12})^t C_{11}\}F_{11}^{-1}f_1$$

$$\bar{d}_1 = -(F_{11}^{-1})^t(d_1 + C_{11}E_{11}^{-1}e_1)$$

$$\bar{d}_2 = d_2 - (F_{11}^{-1}F_{12})^t d_1 + \{C_{12} - C_{11}F_{11}^{-1}F_{12}\}^t E_{11}^{-1}e_1$$

$$\bar{C}_{11} = (E_{11}^{-1})^t C_{11}F_{11}^{-1}$$

$$\bar{C}_{12} = (E_{11}^{-1})^t(C_{11}F_{11}^{-1}F_{12} - C_{12})$$

$$\bar{C}_{21} = -(C_{21} - (E_{11}^{-1}E_{12})^t C_{11})F_{11}^{-1}$$

$$\bar{C}_{22} = C_{22} - (E_{11}^{-1}E_{12})^t C_{12} - C_{21}(F_{11}^{-1}F_{12}) + (E_{11}^{-1}E_{12})^t C_{11}(F_{11}^{-1}F_{12})$$

$$\bar{E}_1 = \begin{pmatrix} I \\ -E_{21} \end{pmatrix} E_{11}^{-1} \quad , \quad \bar{E}_2 = \begin{pmatrix} E_{11}^{-1}E_{12} \\ E_{22} - E_{21}E_{11}^{-1}E_{12} \end{pmatrix}$$

$$\bar{F}_1 = \begin{pmatrix} I \\ -F_{21} \end{pmatrix} F_{11}^{-1} \quad , \quad \bar{F}_2 = \begin{pmatrix} F_{11}^{-1}F_{12} \\ F_{22} - F_{21}F_{11}^{-1}F_{12} \end{pmatrix}$$

$$\bar{e} = \begin{pmatrix} E_{11}^{-1}e_1 \\ e_2 - E_{21}E_{11}^{-1}e_1 \end{pmatrix}, \quad \bar{f} = \begin{pmatrix} F_{11}^{-1}f_1 \\ f_2 - F_{21}F_{11}^{-1}f_1 \end{pmatrix}$$

We will call (2.1.3) the 'canonical form' of the bilinear programming problem (1.3) at $(x^o, y^o)$ . This obviously has the same structure and the same dimensionality as the original problem, so by renaming the variables, we can assume without loss of generality that $(0_m, 0_n)$ is a feasible point in (1.3). Let us assume this.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The bilinear programming problem can thus be exhibited in canonical form for the convenience of future reference

$$(2.1.4)\qquad \begin{cases} \text{maximize} \quad \varphi(x,y) = c^t x + d^t y + x^t C y \\ \text{subject to } Ex \leq e \ , \quad (e \geq 0) \\ \qquad\qquad Fy \leq f \ , \quad (f \geq 0) \\ \qquad\qquad x \geq 0 \ , \quad y \geq 0 \end{cases}$$

The following theorem will partially characterize the desired extreme points.

Theorem 2.1.2. The origin $(0_m, 0_n)$ of the system (2.1.4) is

(i) a Kuhn-Tucker point if $c \leq 0_m$ , $d \leq 0_n$ ;

(ii) a local maximum point if (a) $c \leq 0_m, d \leq 0_n$ and

(b) either $c_i < 0$ or $d_j < 0$ if $c_{ij} > 0$ ;

(iii) a global optimal point if $c \leq 0_m$ $d \leq 0_n$ and $C \leq 0$[†] .

Proof.

(i) By Definition 1.2, $(x,y)$ is a Kuhn-Tucker point if and only if $x \in X_o$ , $y \in Y_o$ and there exist a pair of vectors $u \geq 0$ , $v > 0$ such that

$$(2.1.5)\qquad \begin{gathered} c + Cy - E^t u \leq 0_m \ , \qquad d + C^t x - F^t v \leq 0_n \\ x^t(c + Cy - E^t u) = 0 \ , \quad y^t(d + C^t x - F^t v) = 0 \\ v^t(Ex - e) = 0 \qquad\qquad u^t(Fy - f) = 0 \end{gathered}$$

which is obviously satisfied by $x = 0$ , $y = 0$ , $u = 0$ , $v = 0$ .

---

[†]The condition $C \leq 0$ in (iii) can be replaced by the copositivity of $-C$ (see [C-5]).



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

(ii) Let $x \in R^m$ and $y \in R^n$ be any non-negative vectors. Let $I = \{i | c_i < 0\}$, $J = \{j | d_j < 0\}$ and let $\epsilon$ and $\delta$ be positive parameters. Then

$$\varphi(\epsilon x, \delta y) = \epsilon \cdot c^t x + \delta \cdot d^t y + \epsilon\delta \cdot x^t C y$$

$$= \epsilon \sum_{i \in I} c_i x_i + \delta \sum_{j \in J} d_j y_j + \epsilon\delta \left( \sum_{i \notin I, j \notin J} c_{ij} x_i y_j + \sum_{i \in I \text{ or } j \in J} c_{ij} x_i y_j \right)$$

$$\leq \epsilon \sum_{i \in I} c_i x_i + \delta \sum_{j \in J} d_j y_j + \epsilon\delta \sum_{i \in I \text{ or } j \in J} c_{ij} x_i y_j$$

The inequality follows since $c_{ij} \leq 0$ when $i \notin I$ and $j \notin J$. Obviously the last expression is less than zero for small enough $\epsilon$ and $\delta$ if $I \neq \emptyset$ or $J \neq \emptyset$ since the linear term dominates the quadratic term. This implies that $\varphi(\epsilon x, \delta y) \leq 0 = \varphi(0,0)$ for all $x \geq 0$, $y \geq 0$ and small enough $\epsilon > 0$, $\delta > 0$. If $I = \emptyset$ and $J = \emptyset$, then by assumption $c_{ij} \leq 0$ for all $i,j$ and $\varphi(\epsilon x, \delta y) \leq 0$ for all $x \geq 0$, $y \geq 0$ and for all $\epsilon \geq 0$, $\delta \geq 0$.

(iii) $\varphi(x,y) = c^t x + d^t y + x^t C y \leq 0$ for all $x \geq 0$, $y \geq 0$ since $c \leq 0_m$, $d \leq 0_n$ and $C \leq 0$. Q.E.D.

Statements (i) and (ii) of this theorem can be replaced by stronger ones if the problem is primal non-degenerate.

Theorem 2.1.3. If $e > 0$ and $f > 0$, then the origin $(0_m, 0_n)$ of the system (2.1.4) is

(i) a Kuhn-Tucker point if, and only if $c \leq 0_m$ and $d \leq 0_n$;

(ii) a local maximum if, and only if (a) $c \leq 0_m$, $d \leq 0_n$ and

(b) either $c_i < 0$ or $d_j < 0$ if $c_{ij} > 0$.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Proof. It suffices to prove the only if part.

(i) If $(0_m, 0_n)$ is a Kuhn-Tucker point, then by the third row equations $v^t e = 0$ and $u^t f = 0$. $e > 0$ and $f > 0$ imply $u = 0$, $v = 0$. Hence by the first row equations we have $c \leq 0$, $d \leq 0$.

(ii) Assume $c \not\leq 0$ or $d \not\leq 0$. Then there exists an index $i$ such that $c_i > 0$ or an index $j$ such that $d_j > 0$. Let $c_i > 0$. Since $e > 0$, we can increase $x_i$ up to a certain positive constant $x_i^*$ (possibly $+\infty$) without violating the constraints. This implies that $(0_m, 0_n)$ is not a local maximum, a contradiction to the assumption. Hence the result follows. Q.E.D.

Since these theorems do not fully characterize the extreme points, we are led to introduce the notion of a locally maximum vertex which plays an important role throughout our algorithm.

Definition 2.1.1. Let $s$ be an extreme point of a polyhedral convex set $S$. Then $N_S(s)$ represents the set of the vertices adjacent to $s$ in $S$ and $R_S(s)$ represents the set of points on the rays incident to $s$ in $S$. (If $S$ is bounded, then $R_S(s) = \emptyset$.)

Definition 2.1.2. The pair of extreme points $x^*$ and $y^*$ of the polyhedral convex sets $X_o$ and $Y_o$ is called a locally maximum vertex† if (i) $c \leq 0$, $d \leq 0$ and (ii) $\varphi(x^*, y^*) \geq \varphi(x, y)$ for all $x \in N_{X_o}(x^*) \cup R_{X_o}(x^*)$ and for all $y \in N_{Y_o}(y^*) \cup R_{Y_o}(y^*)$. A locally maximum vertex is called a unique locally maximum vertex if the weak inequality in (ii) is replaced by a strong one.

†In [C-3], A. Charnes and W. W. Cooper used the term 'local star optimum' in place of 'locally maximum vertex'.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

## 2.2 Determination of an Initial Locally Maximum Vertex

In this section we will present an algorithm to determine a locally maximum vertex. As we pointed out in the previous chapter, the bilinear programming problem reduces to a linear programming problem if we temporarily fix $x$ or $y$, and it is quite natural to solve successive linear programs fixing either $x$ or $y$ until we have a 'good' solution.

To be precise, let $(x^0,y^0)$ a pair of any basic feasible solutions of (1.3). Starting from a given pair of basic feasible solutions $(x^i,y^i)$, we will calculate $(x^{i+1},y^{i+1})$ as follows. Let $x^{i+1}$ be an optimal (basic) solution of a linear programming problem

$$\underset{x}{\text{maximize}} \{\varphi(x,y^i) \mid x \in X_o\} \tag{2.2.1}$$

It follows that $\varphi(x^{i+1},y^i) \geq \varphi(x^1,y^i)$ since $x^i \in X_o$. Similarly, let $y^{i+1}$ be an optimal (basic) solution of a linear programming problem

$$\text{maximize } \{\varphi(x^{i+1},y) \mid y \in Y_o\} \tag{2.2.2}$$

Again we have $\varphi(x^{i+1},y^{i+1}) \geq \varphi(x^{i+1},y^i)$ since $y^1 \in Y_o$ and thus $\varphi(x^{i+1},y^{i+1}) \geq \varphi(x^i,y^i)$. This process will terminate in a finite number of steps under either of the following conditions:

(a) there exists a ray incident to $(x^k,y^k)$ along which the objective function can increase without a bound;

(b) $(x^{k+1},y^{k+1}) = (x^k,y^k) \triangleq (x^\infty,y^\infty)$.

If (a) obtains, then there exists an infinite solution and the problem has been "solved". If, on the other hand (b) obtains, then $(x^\infty,y^\infty)$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

gives a Kuhn-Tucker point. Why this is so becomes obvious if we consider the following simplex tableau:

maximize $\{\varphi(x,y^{\infty}) \mid x \in X_o\}$

| | $x_1$ | $x_2$ | $u_1$ | $u_2$ |
|---|---|---|---|---|
| | $c + Cy^{\infty}$ | | 0 | 0 |
| $e_1$ | $E_{11}$ | $E_{12}$ | I | 0 |
| $e_2$ | $E_{21}$ | $E_{22}$ | 0 | I |

Table 2.1. Initial Tableau

| | $x_1$ | $x_2$ | $u_1$ | $u_2$ |
|---|---|---|---|---|
| | 0 | $\bar{c}_2$ | $\bar{c}_1$ | 0 |
| $\bar{e}_1$ | I | $E_{11}^{-1}E_{12}$ | $E_{11}^{-1}$ | 0 |
| $\bar{e}_2$ | 0 | $E_{22} - E_{21}E_{11}^{-1}E_{12}$ | $-E_{21}E_{11}^{-1}$ | I |

Table 2.2. Final Tableau

maximize $\{\varphi(x^{\infty},y) \mid y \in Y_o\}$

| | $y_1$ | $y_2$ | $v_1$ | $v_2$ |
|---|---|---|---|---|
| | $d + C^t x^{\infty}$ | | 0 | 0 |
| $f_1$ | $F_{11}$ | $F_{12}$ | I | 0 |
| $f_2$ | $F_{21}$ | $F_{22}$ | 0 | I |

Table 2.3. Initial Tableau

| | $y_1$ | $y_2$ | $v_1$ | $v_2$ |
|---|---|---|---|---|
| | 0 | $\bar{d}_2$ | $\bar{d}_1$ | 0 |
| $\bar{f}_1$ | I | $F_{11}^{-1}F_{12}$ | $F_{11}^{-1}$ | 0 |
| $\bar{f}_2$ | 0 | $F_{22} - F_{21}F_{11}^{-1}F_{12}$ | $-F_{21}F_{11}^{-1}$ | I |

Table 2.4. Final Tableau

The quantities in the Tables 2.1 ~ 2.4 are given by the expressions following (2.1.3). By the rule of simplex algorithm, if either $(\bar{c}_1^t, \bar{c}_2^t) \not\leq 0$ or $(\bar{d}_1^t, \bar{d}_2^t) \not\leq 0$, then we can increase the objective function by increasing some nonbasic variable if the problem is primal non degenerate, i.e., if $(e_1^t, e_2^t) > 0$, $(f_1^t, f_2^t) > 0$. This is a contradiction to the assumption of the termination and we have established $(\bar{c}_1^t, \bar{c}_2^t) \leq 0$, $(\bar{d}_1^t, \bar{d}_2^t) \leq 0$ if $(e_1^t, e_2^t) > 0$, $(f_1^t, f_2^t) > 0$. By using the standard argument, we can show that there exists a pair of bases for which this condition holds even when the problem is degenerate.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Note that we can start from a feasible tableau when we solve the sub-problems (2.2.1) and (2.2.2). The only difference from the previous steps is their cost rows, which can be calculated with little computation from the final tableau of the previous step. In addition, any efficient special algorithm for a structured linear program is applicable to the subproblems.

Now let us assume that (b) obtains and the problem has been reduced to a canonical form with respect to $(x^{\infty}, y^{\infty})$. The transformed objective function and constraints are readily available from the current tableaux (Table 2.2.2, 2.2.4). Now we can assume that $c \leq 0$ and $d \leq 0$. The origin $(0_m, 0_n)$ of (2.1.4) in its canonical expression is a globally optimal solution if $C \equiv (c_{ij}) \leq 0$ (Theorem 2.1.2). If not let

$$K = \{(i,j) \mid c_{ij} > 0,\ i=1,\dots,m;\ j=1,\dots,n\}, \quad K \neq \emptyset \tag{2.2.3}$$

If we increase $x_i$ and $y_i$ simultaneously for $(i,j) \in K$, then the objective function may be improved. Let

$$\varphi_{ij}(x_i, y_j) = c_i x_i + d_j y_j + c_{ij} x_i y_j, \quad (i,j) \in K \tag{2.2.4}$$

where $c_i \leq 0$, $d_j \leq 0$ and $c_{ij} > 0$.

Proposition 2.2.1. If $\varphi_{ij}(\bar{x}_i, \bar{y}_j) > 0$ for some $\bar{x}_i \geq 0$, $\bar{y}_j \geq 0$, then $\varphi_{ij}(x_i, y_i) > \varphi_{ij}(\bar{x}_i, \bar{y}_j)$ for all $x_i > \bar{x}_i$, $y_j > \bar{y}_j$.

Proof. Let $x_i = \bar{x}_i + \epsilon_i$, $y_j = \bar{y}_j + \epsilon_j$ where $\epsilon_i > 0$, $\epsilon_j > 0$. Note that neither $\bar{x}_i = 0$ nor $\bar{y}_j = 0$ is possible since, say if $\bar{x}_i = 0$,



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

then $\varphi_{ij}(0,\bar{y}_j) = d_j\bar{y}_j \leq 0$ contrary to assumption

$$\varphi_{ij}(x_i,y_j) = c_i(\bar{x}_i + \epsilon_i) + d_j(\bar{y}_j + \epsilon_j) + c_{ij}(\bar{x}_i + \epsilon_i)(\bar{y}_j + \epsilon_j)$$

$$= \varphi_{ij}(\bar{x}_i,\bar{y}_j) + \epsilon_i(c_i + c_{ij}\bar{y}_j) + \epsilon_j(d_j + c_{ij}\bar{x}_i) + c_{ij}\epsilon_i\epsilon_j$$

$$\geq \varphi_{ij}(\bar{x}_i,\bar{y}_j) + \epsilon_i\left(-d_j\frac{\bar{y}_j}{\bar{x}_i}\right) + \epsilon_j\left(-c_i\frac{\bar{x}_i}{\bar{y}_j}\right) + c_{ij}\epsilon_i\epsilon_j > \varphi(\bar{x}_i,\bar{y}_j)$$

where $(c_i + c_{ij}\bar{y}_j) > -d_j\frac{\bar{y}_j}{\bar{x}_i}$ for example follows from $\varphi(\bar{x}_i,\bar{y}_j) > 0$ & $\bar{x}_i > 0$ and the last inequality follows from $d_j \leq 0$, $c_i \leq 0$ and $c_{ij} > 0$ . Q.E.D.

This proposition shows that if we can improve the objective function in the direction of a certain pair of edges defined by $\lambda\bar{x} = (0,\ldots,0, \lambda\bar{x}_i,0,\ldots,0)$ , $\mu\bar{y} = (0,\ldots,0,\mu\bar{y}_j,0,\ldots,0)$ incident to the origin where $\lambda \geq 0$, $\mu \geq 0$ are parameters, then the maximum improvement is achieved by choosing $\lambda$ and $\mu$ so that $\lambda\bar{x}$ and $\mu\bar{y}$ are vertices of the feasible sets $X_o$ and $Y_o$ , respectively. These vertices have the form $x^i = (0,\ldots,0,\bar{x}_i,0,\ldots,0)$ and $y^j = (0,\ldots,0,\bar{y}_j,0,\ldots,0)$ where

$$\bar{x}_i = \min\left\{\frac{e_p}{e_{pi}} \;\middle|\; e_{pi} > 0 \text{ , } p=1,\ldots,k_1\right\}$$

(2.2.5)

$$\bar{y}_j = \min\left\{\frac{f_q}{f_{qj}} \;\middle|\; f_{qj} > 0 \text{ , } q=1,\ldots,k_2\right\}$$

and $e_p$ and $e_{pi}$ represent the $p^{th}$ element of $e$ and $(p,i)^{th}$ element of $E$ , respectively, and so on. Let

$$\varphi_{i_oj_o} = \max\{\varphi_{ij}(\bar{x}_i,\bar{y}_j) \mid (i,j) \in K\}$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Case 1. $\varphi_{i_o j_o} > 0$ . $(x^{i_o}, y^{j_o})$ is the best improved solution among the combinations of the adjacent vertex pairs $x^i, y^j$ . In this case, we will restart the whole suboptimization process from this point.

Case 2. $\varphi_{i_o j_o} \leq 0$ . By definition, $(O_m, O_n)$ is a locally maximum vertex.

In the latter case, there are various alternative ideas we might explore. For example if $\varphi_{i_o j_o} = 0$ we might look for improvements from the locally maximum vertex $x^{i_o}, y^{j_o}$ and continue in this manner until we have exhausted the entire set of vertices that are connected by edge pairs and yield the same objective value. There remains the possibility of cycling among stationary points which requires some additional device (like maintaining the list of vertices) to prevent this phenomenon. Apparently, the standard perturbation technique of linear programming does not extend in any obvious way. However, maintaining a list of stationary points that are connected by edge pairs and yield the same value of the objective function appears to be practical in most applications.

Alternatively, we can quit the suboptimization search just described at any point at which $\varphi_{i_o j_o} \leq 0$ and proceed to the next stage of the algorithm.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

## 3. THE CUTTING PLANES AT A LOCALLY MAXIMUM VERTEX

### 3.1 Introduction

Let us assume that we are now at the $p^{th}$ locally maximum vertex and that this vertex has already been translated into the origin. Let

$$\varphi_o = \max\{\varphi_1, \ldots, \varphi_p\} \tag{3.1.1}$$

where $\varphi_j$ represents the value of the objective function at the $j^{th}$ locally maximum vertex.

In this chapter, we will consider the introduction of a cutting plane which eliminates the current locally maximum vertex but does not eliminate any extreme point $(x,y) \in X_o \times Y_o$ which may achieve a greater objective value than $\varphi_o$. We will call a cut satisfying these requirements a 'legitimate' cut.

The cuts to be considered in the following sections will have the form

$$g^t x + h^t y \geq \tau \tag{3.1.2}$$

where $g \geq 0, h \geq 0$. Obviously, we must have $(g^t, h^t) \neq (0,0)$ and $\tau > 0$ if the cut is to be a legitimate one and our task here is to determine the vectors $g,h$ and a scalar $\tau$. Note that if $g^t$ and $h^t$ are both different from zero the inclusion of the cut destroys the separable structure of the constraints into an X set and a Y set. Our basic strategy throughout this chapter will be stated as follows: Let $\bar{X}_o$ and $\bar{Y}_o$ be any supersets of $X_o$ and $Y_o$ respectively, i.e.,



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$X_o \subset \bar{X}_o$ , $Y_o \subset Y_o$ . Let

(3.1.3) $\Psi(\sigma) = \max\{\varphi(x,y) \mid g^t x + h^t y \leq \sigma\ ,\ x \in \bar{X}_o,\ y \in \bar{Y}_o\}$

where $\sigma$ is a non-negative scalar and $(0_m, 0_n) \leq (g^t, h^t) \neq (0_m, 0_n)$ . If we can show that

(3.1.4) $$\Psi(\sigma) \leq \varphi_o - \varphi_p \quad , \quad 0 \leq \sigma \leq \sigma^*$$

for some positive constant $\sigma^*$ , then the cut (3.1.2) with $\tau = \sigma^*$ is obviously legitimate. Of course, unless $\bar{X}_o$ and $\bar{Y}_o$ are very special sets it would not be practical to solve the parametric (non-concave) quadratic programming problem (3.1.3). So we will proceed to choose $g$ , $h$ , $\bar{X}_o$ and $\bar{Y}_o$ in such a way that

(i) the calculation of $\sigma^*$ does not require an excessive amount of computation and yet in a certain sense

(ii) it generates a 'deep' cut.

Obviously, there is a trade off between these two requirements.

In the next section, we will choose

(3.1.5) $$g > 0\ ,\ h > 0\ ,\ \bar{X}_o = R^m_+\ ,\ \bar{Y}_o = R^n_+$$

This choice makes the task of calculating $\Psi(\sigma)$ easy and a very simple analytical expression for $\tau$ becomes available by elaborating on the special structure of our problem. But this choice of $\bar{X}_o$ and $\bar{Y}_o$ could result in a 'shallower' cut instead of a possible 'deeper' cut. In addition, the choice of the vectors $g > 0$ and $h > 0$ necessarily destroys the fundamental separable structure of the bilinear programming problem. These observations lead us to consider in Section 3.3 the



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

following choice which maintains the separability:

(3.1.6) $g > 0$ , $h = 0$ , $\bar{X}_o = R_+^m$ , $\bar{Y}_o = \{y \in R_+^n | \xi^t y \leq \xi_o\}$

(3.1.7) $g = 0$ , $h > 0$ , $\bar{X}_o = \{x \in R_+^m | \zeta^t x \leq \zeta_o\}$ , $\bar{Y}_o = R_+^n$

(3.1.8) $g > 0$ , $h = 0$ , $\bar{X}_o = R_+^m$ , $\bar{Y}_o = Y_o$

(3.1.9) $g = 0$ , $h > 0$ , $\bar{X}_o = X_o$ , $\bar{Y}_o = R_+^n$

where $\xi$ and $\zeta$ are positive vectors to be determined. Such choices do not destroy the separable structure. The first two lead to easy calculation but could yield shallower cuts. The last two, (3.1.8) and (3.1.9), necessitate the solution of a family of parametric linear programming problems and are less desirable from the computational point of view. However, the cut generated in this way takes into account the local optimality of the objective function in the region $(0_m, y)$, $y \in Y_o$ or $(x, 0_n)$, $x \in X_o$ and hence is expected to yield stronger cuts.

3.2 The Cuts $g^t x + h^t y \geq \tau$

In this section, we will consider the cut based upon the choice (3.1.5). Let $\psi(\sigma)$ be $\Psi(\sigma)$ of (3.1.3) corresponding to this choice, i.e.,

(3.2.1) $\psi(\sigma) = \max\{\varphi(x,y) | \; g^t x + h^t y \leq \sigma \; , \; x \geq 0 \; , \; y \geq 0\}$

or more explicitly, let $\psi(\sigma)$ be the optimal objective value of the following bilinear programming problem:

(3.2.2)
$$\begin{cases} \text{maximize} & \varphi(x,y) = c^t x + d^t y + x^t C y \\ \text{subject to} & g^t x + h^t y \leq \sigma \quad (\sigma \geq 0;\ g > 0,\ h > 0) \\ & x \geq 0 \; , \; y \geq 0 \end{cases}$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Obviously this problem has an optimal solution for any $\sigma \geq 0$ since the constraint set is non-empty and compact.

Theorem 3.2.1. An optimal solution of (3.2.2) is attained either at the origin $(0_m, 0_n)$ or else there exists an optimal solution $(x^*, y^*)$

(i) $g^t x^* + h^t y^* = \sigma$

(ii) both $x^*$ and $y^*$ have at most one positive component.

Proof. Let $(\bar{x}, \bar{y})$ be an optimal solution such that $(\bar{x}, \bar{y}) \neq (0_m, 0_n)$. Assume that $(\bar{x}, \bar{y})$ does not satisfy (i) and/or (ii). We will show the existence of another optimal solution $(x^*, y^*)$ which satisfies (i) and (ii) by constructing it. Let us consider the linear program

$$\underset{y}{\text{maximize}} \{\varphi(\bar{x}, y) \mid h^t y \leq \sigma - g^t \bar{x}, \; y \geq 0\}$$

Applying the basic theorem of linear programming, there exists an optimal solution $y^*$ to this subproblem satisfying (a) $\varphi(\bar{x}, y^*) \geq \varphi(\bar{x}, \bar{y})$ and (b) either $y^* = 0$ or $h^t y^* = \sigma - g^t \bar{x}$ where $y^*$ has at most one positive component. Since $(\bar{x}, \bar{y})$ is an optimal solution of (3.2.2), $\varphi(\bar{x}, \bar{y}) \geq \varphi(\bar{x}, y^*)$. Hence, $(\bar{x}, y^*)$ is also an optimal solution of (3.2.2).

Case 1. $y^* = 0$.

Consider the linear program

$$\underset{x}{\text{maximize}} \{\varphi(x, 0_n) \mid g^t x \leq \sigma, \; x \geq 0\}$$

Again by the basic theorem of linear programming, there exists an optimal solution $x^*$ to this subproblem satisfying (a) $\varphi(x^*, y^*) \geq \varphi(\bar{x}, y^*)$ and (b) either $x^* = 0$ or $g^t x^* = \sigma$ where $x^*$ has at most one positive



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

component. By the optimality of $(\bar{x},y^*)$, $(x^*,y^*)$ is also an optimal solution of (3.2.2). Obviously, $(x^*,y^*)$ satisfies (i) and (ii).

Case 2. $h^t y^* = \sigma - g^t \bar{x}$ .

Consider the linear program

$$\underset{x}{\text{maximize}} \{\varphi(x,y^*) \mid g^t x = \sigma - h^t y^*,\ x \geq 0\}$$

By the same reasoning as before, there exists an optimal solution $x^*$ such that (a) $\varphi(x^*,y^*) \geq \varphi(\bar{x},y^*)$ and (b) $g^t x^* = \sigma - h^t y^*$ and $x^*$ has at most one positive component. Obviously $(x^*,y^*)$ is an optimal solution of (3.2.2) satisfying (i) and (ii). Q.E.D.

Corollary 3.2.1.1. $\psi(\sigma) = \max\left[0,\ \max_{ij}\{\psi_{ij}(\sigma) \mid i=1,\ldots,m;\ j=1,\ldots,n\}\right]$

where

$$(3.2.3)\quad \psi_{ij}(\sigma) = \max_{x_i,y_j}\left\{c_i x_i + d_j y_j + c_{ij} x_i y_j \mid g_i x_i + h_j y_j = \sigma,\ x_i \geq 0,\ y_j \geq 0\right\}$$

Proof. By Theorem 3.2.1, either $(0_m,0_n)$ is an optimal solution or else there exists an optimal solution among the vectors $x=(0,\ldots,0,x_i,0,\ldots,0)$, $y=(0,\ldots,0,y_j,0,\ldots,0)$ for certain $i,j$, satisfying $g_i x_i + h_j y_j = \sigma$, $x_i \geq 0$, $y_j \geq 0$. Evidently $\varphi(0_m,0_n) = 0$ and the result follows. Q.E.D.

Note that $c_i \leq 0$ and $d_j \leq 0$ for all $i,j$ since the origin is a locally maximum vertex by assumption, and it follows that $\psi_{ij}(\sigma) \leq 0$ if $(i,j) \notin K$ where $K = \{(i,j) \mid c_{ij} > 0,\ i=1,\ldots,m;\ j=1,\ldots,n\}$ .



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Lemma 3.2.2. For $(i,j) \in K$ ,

$$(3.2.4) \qquad \psi_{ij}(\sigma) = \begin{cases} 0 & 0 \le \sigma \le \sigma_{ij} \\ \frac{1}{4}\bar{c}_{ij}\sigma^2 + \frac{\bar{c}_i + \bar{d}_j}{2}\sigma + \frac{(\bar{c}_i - \bar{d}_j)^2}{4\bar{c}_{ij}} , & \sigma > \sigma_{ij} \end{cases}$$

where

$$(3.2.5) \qquad \bar{c}_i = \frac{c_i}{g_i} , \quad \bar{d}_j = \frac{d_j}{h_j} , \quad \bar{c}_{ij} = \frac{c_{ij}}{g_i h_j} , \quad \sigma_{ij} = \frac{(\sqrt{-\bar{c}_i} + \sqrt{-\bar{d}_j})^2}{\bar{c}_{ij}}$$

Proof.

$$\psi_{ij}(\sigma) = \max_{x_i, y_j} \left\{ c_i x_i + d_j y_j + c_{ij} x_i y_j \,\middle|\, g_i x_i + h_j y_j = \sigma,\ x_i \ge 0 ,\ y_j \ge 0 \right\}$$

$$= \max_{u,v} \left\{ \bar{c}_i u + \bar{d}_j v + \bar{c}_{ij} uv \,\middle|\, u + v = \sigma ,\ u \ge 0 ,\ v \ge 0 \right\}$$

$$= \max_{u} \left\{ -\bar{c}_{ij} u^2 + (\bar{c}_i - \bar{d}_j + \bar{c}_{ij}\sigma)u + \bar{d}_j\sigma \,\middle|\, 0 \le u \le \sigma \right\}$$

$$= \begin{cases} \max(\bar{c}_i, \bar{d}_j)\sigma & 0 \le \sigma \le \dfrac{|\bar{c}_i - \bar{d}_j|}{\bar{c}_{ij}} \\ \dfrac{\bar{c}_{ij}}{4}\sigma^2 + \dfrac{\bar{c}_i + \bar{d}_j}{2}\sigma + \dfrac{(\bar{c}_i - \bar{d}_j)^2}{4\bar{c}_{ij}} & \sigma \ge \dfrac{|\bar{c}_i - \bar{d}_j|}{\bar{c}_{ij}} \end{cases}$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

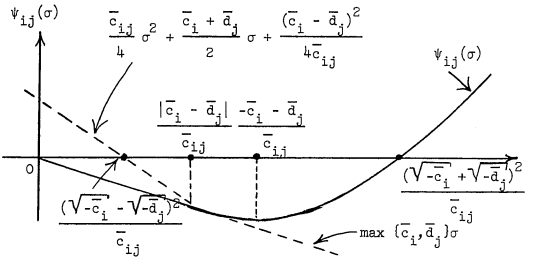



Figure 3.2.1.

Obviously $\max\{c_i, d_j\}\sigma \le 0$ for $\sigma \ge 0$ since $\bar{c}_i \le 0$, $\bar{d}_j \le 0$. Simple calculation shows that

$$\frac{\bar{c}_{ij}}{4}\sigma^2 + \frac{\bar{c}_i + \bar{d}_j}{2}\sigma + \frac{(\bar{c}_i - \bar{d}_j)^2}{4\bar{c}_{ij}} \quad \begin{matrix} \le \\ \ge \end{matrix} \quad 0 \quad \text{if} \quad \begin{matrix} 0 \le \sigma \le \sigma_{ij} \\ \sigma \ge \sigma_{ij} \end{matrix}$$

Moreover

$$\sigma_{ij} \equiv \frac{(\sqrt{-\bar{c}_i} + \sqrt{-\bar{d}_j})^2}{\bar{c}_{ij}} \ge \frac{|\bar{c}_i - \bar{d}_j|}{\bar{c}_{ij}}$$

since $\bar{c}_i \le 0$, $\bar{d}_j \le 0$. Now that Figure 3.2.1 is justified, the result follows. Q.E.D.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Now we will be more specific about the choice of $g$ and $h$ to simplify the expression of $\psi(\sigma)$ .

Let $\hat{g} = \{\hat{g}_i\}_{i=1}^{m}$ and $\hat{h} = \{\hat{h}_j\}_{j=1}^{n}$ be defined as follows ($\hat{g} \in R_+^m$ , $\hat{h} \in R_+^n$)

$$\hat{g}_i = \begin{cases} -c_i & \text{if } c_i < 0 \\ g_o & \text{if } c_i = 0 \end{cases} \tag{3.2.6}$$

$$\hat{h}_j = \begin{cases} -d_j & \text{if } d_j < 0 \\ h_o & \text{if } d_j = 0 \end{cases} \tag{3.2.7}$$

where $g_o$ and $h_o$ are some positive constants.

Let

$$\begin{aligned} I &= \{i \mid c_i < 0 \ , \ i=1,\ldots,m\} \\ J &= \{j \mid d_j < 0 \ , \ j=1,\ldots,n\} \end{aligned} \tag{3.2.8}$$

and $I \times J$ their cartesian product.

Theorem 3.2.3. If we choose $g = \hat{g}$ and $h = \hat{h}$ , then $\psi(\sigma) = \max\{\psi_1(\sigma), \psi_2(\sigma), \psi_3(\sigma)\}$ where

$$\psi_1(\sigma) = \begin{cases} 0 & 0 \le \sigma \le \dfrac{4}{k_{IJ}} \\ \dfrac{k_{IJ}}{4} \sigma^2 - \sigma & \sigma \ge \dfrac{4}{k_{IJ}} \end{cases}$$

$$\psi_2(\sigma) = \begin{cases} 0 & 0 \le \sigma \le \dfrac{1}{k} \\ \dfrac{k}{4} \left(\sigma - \dfrac{1}{k}\right)^2 & \sigma \ge \dfrac{1}{k} \end{cases}$$

$$\psi_3(\sigma) = \frac{k_{IJ}}{4} \sigma^2 \qquad \sigma \ge 0$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

and

$$(3.2.9)\qquad \begin{cases} k_{IJ} = \max\{\bar{c}_{ij} \mid (i,j) \in (I \times J) \cap K\} \\ k_{\bar{I}\bar{J}} = \max\{\bar{c}_{ij} \mid (i,j) \in (\bar{I} \times \bar{J}) \cap K\} \\ k = \max\{\bar{c}_{ij} \mid (i,j) \in \left((\bar{I} \times J) \cup (I \times \bar{J})\right) \cap K\} \end{cases}$$

(By convention, quantities in the expression (3.2.9) are assumed to be $-\infty$ if the defining index set is empty.)

Proof. By Corollary 3.2.1.1, $\psi(\sigma) = \max[0, \max\{\psi_{ij}(\sigma) \mid (i,j) \in K\}]$.

Case (i) $(i,j) \in (I \times J) \cap K$. In this case $\bar{c}_i = \bar{d}_j = -1$. Substituting this into (3.2.4) we have

$$\psi_{ij}(\sigma) = \begin{cases} 0 & 0 \le \sigma \le \dfrac{4}{\bar{c}_{ij}} \\ \dfrac{\bar{c}_{ij}}{4}\sigma^2 - \sigma & \sigma \ge \dfrac{4}{\bar{c}_{ij}} \end{cases}$$

Obviously $\bar{c}_{i_1j_1} \ge \bar{c}_{i_2j_2}$ implies $\psi_{i_1j_1}(\sigma) \ge \psi_{i_2j_2}(\sigma)$ for all $\sigma \ge 0$, so that $\max\{\psi_{ij}(\sigma) \mid (i,j) \in (I \times J) \cap K\} = \psi_1(\sigma)$.

Case (ii) $(i,j) \in \left((\bar{I} \times J) \cup (I \times \bar{J})\right) \cap K$. In this case either $\bar{c}_i = 0$, $\bar{d}_j = -1$ or $\bar{c}_i = -1$, $\bar{d}_j = 0$. Substituting these into (3.2.4), we have

$$\psi_{ij}(\sigma) = \begin{cases} 0 & 0 \le \sigma \le \dfrac{1}{\bar{c}_{ij}} \\ \dfrac{\bar{c}_{ij}}{4}\left(\sigma - \dfrac{1}{\bar{c}_{ij}}\right)^2 & \sigma \ge \dfrac{1}{\bar{c}_{ij}} \end{cases}$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Again it is obvious that if $\bar{c}_{i_2 j_2} \geq \bar{c}_{i_2 j_2}$ , then $\psi_{i_1 j_1}(\sigma) \geq \psi_{i_2 j_2}(\sigma)$ and

$$\max_i \{\psi_{ij}(\sigma) \mid (i,j) \in ((\bar{I} \times J) \cup (I \times \bar{J})) \cap K\} = \psi_2(\sigma)$$

Case (iii). $\underline{(i,j) \in (\bar{I} \times \bar{J}) \cap K}$. In this case $\bar{c}_i = \bar{d}_j = 0$ and by (3.2.4), we have

$$\max \{\psi_{ij}(\sigma) \mid (i,j) \in (\bar{I} \times \bar{J}) \cap K\} = \psi_3(\sigma) \qquad \text{Q.E.D.}$$

Theorem 3.2.3 shows that the only thing required to obtain $\psi(\sigma)$ when $g = \hat{g}$ , $h = \hat{h}$ is the comparison of the various $\bar{c}_{ij} = c_{ij}/g_i h_j$ with each other. Hence

$$\sigma^* = \max \{\sigma' \mid \psi(\sigma) \leq \varphi_o - \varphi_p \ , \ \sigma \in [0,\sigma']\}$$

can be calculated by solving (at most) three quadratic equations.

Example 1. $\varphi(x_1,x_2,y_1,y_2) = -x_1 - y_2 + x_1 y_2 + \frac{1}{8} x_2 y_1 + \frac{1}{2} x_2 y_2$ ;

$\varphi_o - \varphi_p = 3/2$ ; $I = \{1\}$ , $J = \{2\}$ , $K = \{1,2\}$

$k_{IJ} = 1$ , $k_{\bar{I}\bar{J}} = 1/8$ , $k = 1/2$ $(g_o = h_o = 1)$; $\sigma^* = 2+\sqrt{10}$

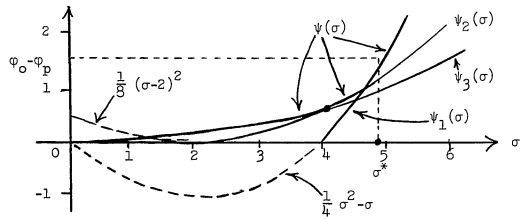



Figure 3.2.2



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Example 2. $\varphi(x_1,x_2,y_1,y_2) = -x_1 - y_1 - y_2 + 2x_1y_1 - x_1y_2 + \frac{1}{3}x_2y_1 + \frac{1}{4}x_2y_2$;

$\varphi_o - \varphi_p = 3/2$ ; $I = \{1\}$ , $J = \{1,2\}$ , $K = \{1,2\}$

$k_{IJ} = 2$ , $k_{\bar{I}\bar{J}} = -\infty$ , $k = 1/3\ (g_o - 1)$ ; $\sigma^* = 3$ .

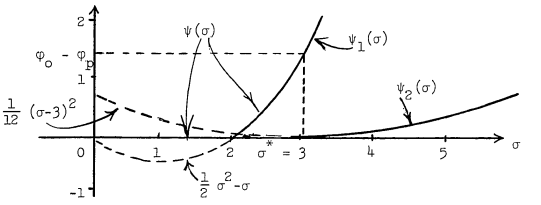



Figure 3.2.3.

It will be apparent from the examples that if $k_{\bar{I}\bar{J}} > -\infty$ , i.e., if $(\bar{I} \times \bar{J}) \cap K \neq \emptyset$ then $\psi(\sigma) = \psi_3(\sigma)$ for small enough $\sigma > 0$ and $\sigma^* = 0$ if $\varphi_o - \varphi_p = 0$ . So the constraint to be adjoined will not , in general, be a legitimate cut.

Theorem 3.2.4. Let $g = \hat{g}$ and $h = \hat{h}$ . Then $\sigma^* > 0$ if either of the following two conditions holds:

(i) $(\bar{I} \times \bar{J}) \cap K = \emptyset$

(ii) $\varphi_o - \varphi_p > 0$

Proof.

(i) If $(\bar{I} \times \bar{J}) \cap K = \emptyset$ , then $\psi(\sigma) = \max\{0, \psi_1(\sigma), \psi_2(\sigma)\}$ .



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

By Theorem 3.2.3, $\psi_1(\sigma)$ and $\psi_2(\sigma)$ are zero up to certain positive constants. Hence $\sigma^* > 0$ .

(ii) If $\varphi_o - \varphi_p > 0$ , then it is obvious from the expressions of $\psi_1(\sigma)$ , $\psi_2(\sigma)$ and $\psi_3(\sigma)$ that $\psi(\sigma) = \varphi_o - \varphi_p$ gives a strictly positive root, so that $\sigma^* > 0$ . Q.E.D.

Corollary 3.2.4.1. Let $g = \hat{g}$ and $h = \hat{h}$ . Then $\sigma^* > 0$ if either $c < 0$ or $d < 0$ .

Proof. In this case, either $\bar{I} = \emptyset$ or $\bar{J} = \emptyset$ and condition (i) of Theorem 3.2.4 holds. Q.E.D.

Corollary 3.2.4.2. Let $g = \hat{g}$ , $h = \hat{h}$ . Then $\sigma^* > 0$ if $e > 0$ and $f > 0$ .

Proof. It suffices to show that $(\bar{I} \times \bar{J}) \cap K = \emptyset$ . Assume not. Then there exists a pair of indices $i,j$ for which $c_i = d_j = 0$ and $c_{ij} > 0$ . Hence the objective function will be improved if we can increase $x_i$ and $y_j$ without violating the feasibility. But we can do this up to some positive constants $x_i^*, y_j^*$ since $e > 0$ , $f > 0$ . This is a contradiction of the assumption that the current origin is a locally maximum vertex. Q.E.D.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

3.3 The Cuts $g^t x \geq \tau_x$ , $h^t y \geq \tau_y$ .

In this section, we will consider the constraints of the form $g^t x \geq \tau_x$ or $h^t y \geq \tau_y$ , based upon the choice (3.1.6) ~ (3.1.9). Since everything is symmetric with respect to $X$ and $Y$ , most of the results will be proved relative to the cuts in $X$ space only, i.e., the cuts $g^t x \geq \tau_x$ .

First we will consider (3.1.8)

$$(3.3.1) \qquad \begin{cases} \text{maximize} \quad \varphi(x,y) = c^t x + d^t y + x^t C y \\ \text{subject to} \quad g^t x \leq \sigma \ , \ x \geq 0 \ ; \ (\sigma \geq 0;\ g > 0) \\ \qquad\qquad\quad y \in Y_o = \{y \mid Fy \leq f \ , \ y \geq 0\} \end{cases}$$

and let $\psi_x(\sigma)$ be the optimal objective value of this program. As we stated before, if $\psi_x(\sigma) \leq \varphi_o - \varphi_p$ for $\sigma \in [0, \sigma_x^*]$ for some positive constant $\sigma_x^*$ , then $g^t x \geq \sigma_x^*$ is a legitimate cut provided $0_m \neq g \geq 0_m$ . Now let us proceed to determine $\psi_x(\sigma)$ .

Theorem 3.3.1. For any fixed $y \in Y_o$ , the maximum of (3.3.1) is attained either at $x = 0$ or else there exists an optimal solution $x^*$ such that $g^t x^* = \sigma$ and $x^*$ has at most one positive component.

Proof. This is analogous to the proof of Theorem 3.2.1 and is omitted here.

Let $\bar{c}_i = c_i / g_i$ , and let $c_{i*}$ and $\bar{c}_{i*}$ be the n-dimensional vectors with $c_{ij}$ and $c_{ij}/g_i$ as their $j^{th}$ components.

Theorem 3.3.2.

$$(3.3.2) \qquad \psi_x(\sigma) = \max \left[ 0, \ \max_i \{ \bar{c}_i \sigma + L_{x,i}(\sigma) \mid i = 1, \ldots, m \} \right]$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

where

$$(3.3.3)\qquad L_{x,i}(\sigma) = \max\{\ell_{x,i}(\sigma) = (d + \sigma\bar{c}_{i*})^t y \mid Fy \le f\ ,\ y \ge 0\}$$

Proof. For any fixed $y \in Y_o$, (3.3.1) has an optimal solution $x = 0$ or else there exists an optimal solution among the vectors $x^i = (0,\ldots,0,\sigma/g_i,0,\ldots,0)$, $i=1,\ldots,m$ (see Theorem 3.3.1). Hence

$$\begin{aligned}
\psi_x(\sigma) &= \max_{x,y}\{c^t x + d^t y + x^t C y \mid g^t x \le \sigma\ ,\ x \ge 0\ ,\ y \in Y_o\} \\
&= \max_{y}\left[\max_{x}\{c^t x + d^t y + x^t C y \mid x=0 \text{ or } x=x^i,\ i=1,\ldots,m\} \mid y \in Y_o\right] \\
&= \max_{y}\left[\max\left\{d^t y,\ \max_i\{\bar{c}_i\sigma + d^t y + \sigma\bar{c}_{i*}^t y \mid i=1,\ldots,m\}\right\} \,\middle|\, y \in Y_o\right] \\
&= \max\left[0,\ \max_i\left\{\bar{c}_i\sigma + \max_y\{(d + \sigma\bar{c}_{i*})^t y \mid y \in Y_o\} \mid i=1,\ldots,m\right\}\right] \\
&= \max\left[0,\ \max_i\{\bar{c}_i\sigma + L_{x,i}(\sigma) \mid i=1,\ldots,m\}\right] \qquad \text{Q.E.D.}
\end{aligned}$$

Let

$$(3.3.4)\qquad \sigma_x^* = \max\{\sigma' \mid \psi_x(\sigma) \le \varphi_o - \varphi_p \quad \text{for all} \quad \sigma \in [0,\sigma']\}$$

then, by Theorem 3.3.2,

$$(3.3.5)\qquad \sigma_x^* = \min_i\{\sigma_{x,i}^* \mid i=1,\ldots,m\}$$

where

$$(3.3.6)\qquad \sigma_{x,i}^* = \max\{\sigma' \mid \bar{c}_i\sigma + L_{x,i}(\sigma) \le \varphi_o - \varphi_p \quad \text{for all} \quad \sigma \in [0,\sigma']\}$$

and our task of determining $\sigma_x^*$ reduces to that of solving n parametric



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

linear programs

$$(3.3.7) \qquad \begin{cases} \text{maximize} & \ell_{x,i}(\sigma) = (d + \sigma \bar{c}_{i*})^t y \\ \text{subject to} & Fy \leq f \\ & y \geq 0 \end{cases}$$

Note that $f \geq 0$ and $d \leq 0$, so that $y = 0$ is an optimal solution of this program for all $i$ when $\sigma = 0$ and it is expected that only a few pivotal operations will be required before we find $\sigma^*_{x,i}$ provided $\varphi_o - \varphi_p$ is not too large. Further, note that if $\bar{c}_{i*} \leq 0$, then $\ell_{x,i}(\sigma) \leq 0$ for all $y \geq 0$. This implies that $\sigma^*_{x,i} = +\infty$ if $\bar{c}_{i*} \leq 0$ and the corresponding subproblems can be ignored. Obviously, the efficiency of the process used to determine $\sigma^*_x$ will be affected by the order in which these (at most) $n$ parametric linear programs are solved, but there seems to be no a priori criterion of a 'good' ordering.

Let us illustrate $\psi_x(\sigma)$ and $\sigma^*_x$ by an example.

Example

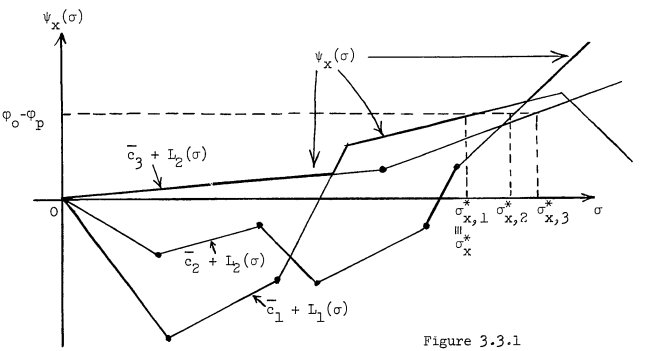



Figure 3.3.1



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Theorem 3.3.3. If $d < 0$ , then $\sigma_x^* \geq \min\{-d_j/\bar{c}_{ij} | \bar{c}_{ij} > 0 , i=1,\ldots,m, j=1,\ldots,n\}$.

Proof. If $d < 0$ , then $\ell_i(\sigma) < 0$ for $\sigma < \sigma_i$ where $\sigma_i = \min\{-d_i/c_{ij} | c_{ij} \geq 0 , j=1,\ldots,n\} > 0$ so that $L_i(\sigma) \leq 0$ up to $\sigma_i$ . Thus $\psi_x(\sigma) \leq 0$ for $\sigma \leq \min\{\sigma_i | i=1,\ldots,n\}$ and the result follows. Q.E.D.

Theorem 3.3.4. Assume that $Y_o$ is a bounded set in $R^n$ . Then $\sigma_x^* > 0$ provided $\varphi_o - \varphi_p > 0$ . More precisely, if $\varphi_o - \varphi_p \geq \epsilon$ , then $\sigma_x^* \geq \epsilon/\alpha K_x$ where $K_x = \max\{c_{ij}/g_i | i=1,\ldots,m; j=1,\ldots,n\}$ and $\alpha = \max\left\{\sum_{j=1}^{n} y_j | y \in Y_o\right\}$ .

Proof. We will prove the latter (stronger) statement.

Assume $\varphi_o - \varphi_p \geq \epsilon$ . It suffices to prove $\sigma_{x,i}^* \geq \epsilon/\alpha K_x$ for all $i=1,\ldots,m$ . By definition

$$\sigma_{x,i}^* = \max\{\sigma' | R_i(\sigma) \leq \varphi_o - \varphi_p ; \ \forall \sigma \in [0,\sigma']\}$$

where

$$R_i(\sigma) = \bar{c}_i\sigma + \max\{(d + \sigma\bar{c}_{i*})^t y \mid y \in Y_o\}$$

Obviously, if $R_i'(\sigma) \geq R_i(\sigma)$ for all $\sigma \geq 0$ , then

$$\sigma_{x,i}^* \geq \lambda_i \equiv \max\{\sigma' | R_i'(\sigma) \leq \varphi_o - \varphi_p ; \ \forall \sigma \in [0,\sigma']\}$$

Let

$$R_i'(\sigma) = \max\{\sigma\bar{c}_{i*}^t y | y \in Y_o\}$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

which is always greater than $R_i(\sigma)$ since $\bar{c}_i \leq 0$ , $d \leq 0$

$$\lambda_i = \max\left\{\sigma' \mid \max\{\sigma \bar{c}_i^t y \mid y \in Y_o\} \leq \varphi_o - \varphi_p \ ; \ \forall\, \sigma \in [0,\sigma']\right\}$$

$$\geq \max\left\{\sigma' \mid \sigma K_x \max\left\{\sum_{j=1}^{n} y_i \,\middle|\, y \in Y_o\right\} \leq \varphi_o - \varphi_p \ ; \ \forall\, \sigma \in [0,\sigma']\right\}$$

$$= \max\{\sigma' \mid \alpha K_x \sigma \leq \varphi_o - \varphi_p \ ; \ \forall\, \sigma \in [0,\sigma']\}$$

$$= \frac{\varphi_o - \varphi_p}{\alpha K_x} \geq \frac{\epsilon}{\alpha K_x}$$

Thus $\sigma^*_{x,i} \geq \epsilon/\alpha K_x$ and the result has been established. Q.E.D.

Up to this point, $g$ could be any positive vector. Let us now be more specific about the choice of this vector.

Our first choice will be $g = \hat{g}$ where $\hat{g}$ is defined by (3.2.6), i.e.,

$$g_i = \begin{cases} -c_i & \text{if} \quad c_i < 0 \\ g_o & \text{if} \quad c_i = 0 \end{cases} \tag{3.3.8}$$

If $c < 0$ , then $g = -c$ and $g^t x = \tau$ is parallel to $\varphi(x,y)$ at $y = 0$.

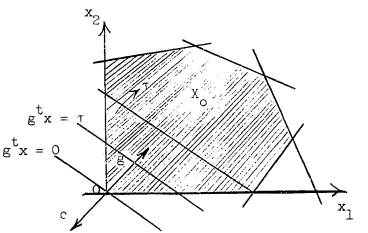



Figure 3.3.2



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Unfortunately, this choice does not simplify the calculation of $\sigma_x^*$ as much as it did in the preceding section. However, it will help us to obtain an upper bound for $\varphi(x,y)$ . This will be discussed further in the next section.

The following theorem leads us to the second choice of $g$ .

Theorem 3.3.5. If $e > 0$ , i.e., if the origin of the X space is a non-degenerate vertex, then $\bar{g}^t x \geq \max\{1, \bar{\sigma}_x^*\}$ is a legitimate cut where $\bar{g} = \{\bar{g}_i\}_{i=1}^m$ is defined by

$$\bar{g}_i = \max\left\{ \frac{e_{p_i}}{e_p} \;\middle|\; e_{p_i} > 0 \;,\; p=1,\ldots,k_1 \right\} \;,\; i=1,\ldots,m \tag{3.3.9}$$

and $\bar{\sigma}_x^*$ is the value of $\sigma_x^*$ for the choice of $g = \bar{g}$ .

Proof. For $x = 0_m$, $\varphi(x,y) = d^t y \leq 0$ for all $y \in Y_o$ , so that the origin of the X space cannot generate a strictly better solution $(0_m, y)$ for whatever choice of $y \in Y_o$ . By Theorem 2.1.1, there exists an optimal solution $(x^*, y^*)$ such that $x^* \in \text{ext } X_o$ and $y^* \in \text{ext } Y_o$ . Hence the cutting plane passing through m vertices adjacent to the origin is a legitimate cut since it does not eliminate any vertex other than the origin. Of course, $x^i$ , the vertex adjacent to the origin in the direction of $x_i$ , has the form $x^i = (0,\ldots,0,\bar{x}_i,0,\ldots 0)$ where

$$\bar{x}_i = \min\left\{ \frac{e_p}{e_{p_i}} \;\middle|\; e_{p_i} > 0 \;;\; p=1,\ldots,k_1 \right\} \quad i=1,\ldots,m$$

Obviously the hyperplane passing through these m vertices is given by $\sum_{i=1}^m x_i/\bar{x}_i = 1$ and the result follows. Q.E.D.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The figure below illustrates the hyperplane $\bar{g}^t x = \tau$ for the two-dimensional case.

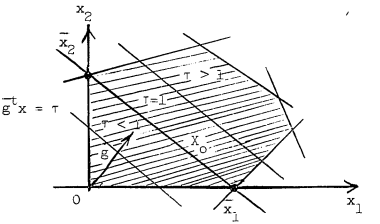



Figure 3.3.3

For completeness, we will state the analogs of Theorems 3.3.2 ~ 3.3.6 relative to the cuts in Y space omitting the proof.

Theorem 3.3.2'.

$$\psi_y(\sigma) = \max\left[0,\ \max_j \{\bar{d}_j\sigma + L_{y,j}(\sigma) \mid j=1,\ldots,n\}\right]$$

where

$$L_{y,j}(\sigma) = \max\{\ell_{y,j}(\sigma) = (c + \sigma\bar{c}_{*j})^t x \mid Ex \leq e\ ,\ x \geq 0\}$$

where $\bar{d}_j = d_j/h_j$ and $\bar{c}_{*j}$ is an m dimensional vector with $c_{ij}/h_j$ as its $i^{th}$ component.

Theorem 3.3.3'. If $c < 0$ , then $\sigma_y^* \geq \min\{-c_i h_j/c_{ij} \mid c_{ij} > 0,$ $i=1,\ldots,m;\ j=1,\ldots,n\}$ .



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Theorem 3.3.4'. Assume that $X_o$ is a bounded set in $R^n$. Then $\sigma_y^* > 0$ provided $\varphi_o - \varphi_p > 0$. More precisely, if $\varphi_o - \varphi_p > \epsilon$, then $\sigma_y^* \geq \epsilon/\beta K_y$ where $K_y = \max\{c_{ij}/h_j \mid i=1,\ldots,m;\ j=1,\ldots,n\}$ and $\beta = \max\left\{\sum_{i=1}^{m} x_i \mid x \in X_o\right\}$.

Theorem 3.3.5'. If $f > 0$, i.e., if the origin of $Y$ space is a non-degenerate vertex, then $\bar{h}_y^t \geq \max\{1, \bar{\sigma}_y^*\}$ is a legitimate cut where $\bar{h} = \{\bar{h}_j\}_{j=1}^{n}$ is defined by

$$\bar{h}_j = \max\left\{\frac{f_{qj}}{f_q} \;\middle|\; f_{qj} > 0,\ q=1,\ldots,k_2\right\};\ j=1,\ldots,n \tag{3.3.10}$$

and $\bar{\sigma}_y^*$ is the value of $\sigma_y^*$ for the choice of $h = \bar{h}$.

It will be observed that the cut based upon (3.3.1) makes use of more information on the behavior of $\varphi(x,y)$ on $Y_o$ than the one based upon (3.2.2), so that it is expected to produce a 'stronger' cut. On the other hand, the exact calculation of $\sigma_x^*$ (or $\sigma_y^*$) will require considerably more computation than the calculation of $\sigma^*$ since we have to solve n (or m) parametric linear programs. It is the author's opinion that this is worthwhile doing in general, but the weaker cuts may work as well in some cases. So we will next consider a cut $g^t x \geq \tau_x$ or $h^t y \geq t_y$ which is weaker but easier to construct.

Let us assume that $Y_o$ is bounded. We will consider the choice (3.1.6)

$$g > 0,\ h = 0,\ X_o = R_+^m,\quad \bar{Y}_o = \{y \in R_+^n \mid \xi^t y \leq \xi_o\} \tag{3.3.11}$$

where $\xi > 0$ is a vector to be determined and $\xi_o = \max\{\xi^t y \mid y \in Y_o\} < \infty$.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

In this case we have to solve

$$(3.3.12) \qquad \begin{cases} \text{maximize} & \varphi(x,y) = c^t x + d^t y + x^t X y \\ \text{subject to} & g^t x \le \sigma \ , \quad x \ge 0 \quad (\sigma \ge 0 \ , \ g > 0) \\ & \xi^t y \le \xi_o \quad y \ge 0 \quad (\xi > 0) \end{cases}$$

Let $\Phi_x(\sigma)$ be the maximum objective value of this program.

Theorem 3.3.6.

$$(3.3.13) \quad \Phi_x(\sigma) = \max\left[0, \max_{i,j} \{\bar{c}_i\sigma + \bar{d}_j\xi_o + \bar{c}_{ij}\sigma\xi_o \mid i=1,\ldots,m;\ j=1,\ldots,n\}\right]$$

where $\bar{c}_i = c_i/g_i$ , $\bar{d}_j = d_j/\xi_j$ , $\bar{c}_{ij} = c_{ij}/g_i\xi_j$ .

Proof. By Theorem 2.1.1, an optimal solution is attained at a pair of extreme points of $X_\sigma = \{x \mid g^t x \le \sigma \ , \ x \ge 0\}$ and $Y_{\xi_o} = \{y \mid \xi^t y \le \xi_o, \ y \ge 0\}$. Obviously, ext $X_\sigma = \{0, x^1, \ldots, x^m\}$ where $x^i = (0,\ldots,0,\bar{c}_i\sigma,0,\ldots,0)$ , $i=1,\ldots,m$ and ext $Y_{\xi_o} = \{0, y^1, \ldots, y^n\}$ where $y^j = (0,\ldots,0,\bar{d}_j\xi_o,0,\ldots,0)$ $j=1,\ldots,n$ . If $x = 0$ , then $\varphi(x,y) = d^t y \le 0$ for all $y \in Y_{\xi_o}$ and $\max\{\varphi(x,y) \mid x=0 \ , \ y \in Y_{\xi_o}\} = 0$ . Similarly, $\max\{\varphi(x,y) \mid y = 0, \ x \in X_o\} = 0$. Now the result follows directly from this and the fact that $\varphi(x^i, y^j) = \bar{c}_i\sigma + \bar{d}_j\xi + \bar{c}_{ij}\sigma\xi$ . Q.E.D.

The determination of

$$(3.3.14) \qquad \sigma'_x = \max\{\sigma' \mid \Phi_x(\sigma) \le \varphi_o - \varphi_p \ ; \ \forall\, \sigma \in [0,\sigma']\}$$

will be much easier than that of $\sigma^*_x$ especially when we take $g = \hat{g}$ and $\xi = \hat{n}$ since in this case $\bar{c}_i = 0$ or $-1$ and $\bar{d}_j = 0$ or $-1$ .



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Theorem 3.3.7. If $Y_o$ is bounded and $d < 0$, then $g^t x \geq \sigma_x^o$ is a legitimate cut where

$$\sigma_x^o = \min\left\{ \frac{-\bar{d}_j}{\bar{c}_{ij}} \;\middle|\; \bar{c}_{ij} > 0 \text{ , } i=1,\ldots,m;\ j=1,\ldots,n \right\} \tag{3.3.15}$$

Proof. By (3.3.13), $\Phi_x(\sigma) \leq 0$ for $\sigma \leq \sigma_x^o$. $\sigma_x^o > 0$ since $d < 0$ and the result follows. Q.E.D.

A similar result holds for the cut in $Y$ space.

Theorem 3.3.7'. If $X_o$ is bounded and $c < 0$, then $h^t y \geq \sigma_y^o$ is a legitimate cut where

$$\sigma_y^o = \min\left\{ \frac{-\bar{c}_i}{\bar{c}_{ij}} \;\middle|\; \bar{c}_{ij} > 0 \text{ , } i=1,\ldots,m;\ j=1,\ldots,n \right\} \tag{3.3.16}$$

Proof. Omitted.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

# 4. SUCCESSIVE LOCALLY MAXIMUM VERTICES AND THE FINITE CONVERGENCE OF THE ALGORITHM TO AN EXTREME $\epsilon$-OPTIMAL SOLUTION

## 4.1 Calculation of Successive Locally Maximum Vertex

In this section, we will discuss an algorithm to determine the $(p+1)$-st locally maximum vertex starting from the $p^{th}$ locally maximum vertex and we will give an upper bound for the objective function $\varphi(x,y)$.

After the addition of a set of new constraints constructed at the $p^{th}$ locally maximum vertex, we have the following augmented system:

$$
(4.1.1)\quad
\begin{cases}
\underset{x,y}{\text{maximize}} \quad \varphi(x,y) = c^t x + d^t y + x^t C y \quad (c \le 0,\ d \le 0) & \\
\text{subject to} \quad Ex \le e \ , \quad (e \ge 0) & \cdots\cdots(1) \\
\qquad\quad Fy \le f \ , \quad (f \ge 0) & \cdots\cdots(2) \\
Gx \ge t_x & \cdots\cdots(3) \\
\qquad Hy \ge t_y & \cdots\cdots(4) \\
Gx + Hy \ge t & \cdots\cdots(5) \\
g^t x \ge \tau_x & \cdots\cdots(6) \\
\qquad h^t y \ge \tau_y & \cdots\cdots(7) \\
g^t x + h^t y \ge \tau & \cdots\cdots(8) \\
x \ge 0 \ , \ y \ge 0 & \cdots\cdots(9)
\end{cases}
$$

where (1), (2), (9) are the original constraints, (3), (4), (5) are the constraints corresponding to the ones adjoined at p-1 previous locally



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

maximum vertices and (6), (7), (8) are the set of newly adjoined constraints at the current ($p^{th}$) locally maximum vertex.

Let

$$\begin{cases} \alpha_x = g^t x^{o,p+1} \equiv \max\limits_{x} \{g^t x \mid x \text{ satisfies (1), (3) and (9)}\} \\ \alpha_y = h^t y^{o,p+1} \equiv \max\limits_{y} \{h^t y \mid y \text{ satisfies (2), (4) and (9)}\} \\ \alpha = g^t x' + h^t y' \equiv \max\limits_{x,y} \{g^t x + h^t y \mid x \text{ and } y \text{ satisfy (1),(2),(3),(4),(5),(9)}\} \end{cases} \tag{4.1.2}$$

The following theorem gives an optimality criterion.

Theorem 4.1.1. If either (i) $\alpha_x < \tau_x$ , (ii) $\alpha_y < \tau_y$ or (iii) $\alpha < \tau$ , then

$$\max\{\varphi(x,y) \mid x \in X_o \text{ , } y \in Y_o\} = \varphi_o \quad \text{where} \quad \varphi_o = \max\{\varphi_1, \ldots, \varphi_p\} \text{ .}$$

Proof. By the construction of the cutting planes, any point $(x,y) \in X_o \times Y_o$ such that $\varphi(x,y) \geq \varphi_o - \varphi_p$ must satisfy (6), (7), and (8) and the result follows. Q.E.D.

Let us assume now that $\alpha_x \geq \tau_x$ , $\alpha_y \geq \tau_y$ , $\alpha \geq \tau$ since if otherwise, then by Theorem 4.1.1 the problem has been solved. Let

$$\begin{cases} X_p = \{x \mid Ex \leq e \text{ , } Gx \geq t_x \text{ , } g^t x \geq \tau_x \text{ , } x \geq 0\} \\ \\ Y_p = \{y \mid Fy \leq f \text{ , } Hy \geq t_y \text{ , } h^t y \geq \tau_y \text{ , } y \geq 0\} \end{cases} \tag{4.1.3}$$

We will apply the suboptimization procedure described in Section 2.2 relative to $X_p$ and $Y_p$ starting from $(x^{o,p+1}, y^{op+1})$ as defined by



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

(4.1.2). To be more precise, we will compute $(x^{i+1},y^{i+1})$ given $(x^i,y^i)$ as follows:

$$x^{i+1} = \arg\max \{\varphi(x,y^i) \mid x \in X_p\}$$
$$y^{i+1} = \arg\max \{\varphi(x^{i+1},y) \mid y \in Y_p\} \tag{4.1.4}$$

and proceed in this manner until we obtain condition (a) or (b) of p.22. If (a) holds then we have an unbounded solution. If, on the other hand, (b) holds, then $(x^\infty,y^\infty)$ gives the $(p+1)$-st locally maximum vertex.

Remark 1. (Updating old cuts using new $\varphi_o$)

It will be obvious from the definition that stronger cuts, i.e., the cuts which eliminate more feasible regions, would have been generated at previous locally maximum vertices if $\varphi_o$ was replaced by some larger constant (see (3.1.3), (3.1.4)). Hence, if the objective value $\varphi_p$ at the $p^{th}$ pseudo vertexwise local maximum is considerably larger than $\max\{\varphi_1,\ldots,\varphi_{p-1}\}$, then it will be worthwhile considering the updating of the old cuts using updated $\varphi_o(=\varphi_p)$. Theoretically, this can be done without any difficulty if we maintain the functional forms of $\psi(\sigma)$, $\psi_x(\sigma)$ and $\psi_y(\sigma)$ associated with each locally maximum vertex for $\sigma$ up to some positive constant. $\psi(\sigma)$ is essentially represented by three constants and is very easy to store, while $\psi_x(\sigma)$ and $\psi_y(\sigma)$ are picewise linear functions having many kinks so that storing these functions will be somewhat more troublesome if there are many local maxima.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Remark 2. (Elimination of redundant constraints)

There is a possibility that some of the constraints defining $X_p$ or $Y_p$ become redundant by the addition of a new cut especially when this cut is expected to be a very deep one. In this case, we can apply, if we like, the techniques developed in [T-1], [E-2], to eliminate these redundant constraints.

Remark 3. (An alternative suboptimization strategy)

An alternative strategy will be to work in $(X_o, Y_o)$ rather than $(X_p, Y_p)$ so long as the cycling among the set of locally maximum vertices does not occur. This will be a reasonable strategy since:

(i) there exists an optimal solution $(x^*, y^*)$ such that $x^* \in \text{ext } X_o$ and $y^* \in \text{ext } Y_o$ (Theorem 2.1.1) and any non-extremal points of $X_o$ and $Y_o$ are inessential;

(ii) $X_o$ and $Y_o$ have fewer constraints and are easier to handle than $X_p$ and $Y_p$;

(iii) if $X_o$ and/or $Y_o$ have some special structure (e.g., unimodularity, Leontief structure) then working in $(X_o, Y_o)$ is much more efficient than working in $(X_p, Y_p)$;

(iv) $(x^{o,p+1}, y^{o,p+1})$ is, in a sense, the farthest point from the $p^{th}$ locally maximum vertex and there is little chance to return to the starting point (i.e., the $p^{th}$ locally maximum vertex) if there exist many locally maximum vertices.

It must be emphasized that we do not use the coupling constraints (5) and (8) throughout this suboptimization procedure since if either $x^\infty \notin \text{ext } X_p$ or $y^\infty \notin \text{ext } Y_p$ then the canonical form relative to



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$(x^{\infty}, y^{\infty})$ will no longer have the fundamental separable structure of the bilinear programming problem. These coupling constraints are maintained only to test the feasibility of the augmented system (4.1.1) or equivalently to test the optimality of the current solution.

Before we state the algorithm explicitly and prove its finite convergence to an extreme $\epsilon$-optimal solution, we will give an upper bound for the objective function $\varphi$ on $X_o \times Y_o$.

Theorem 4.1.2. Let $\varphi_*$ be an optimal objective value for the bilinear programming problem (1.3). If $g = \hat{g}$, $h = \hat{h}$, then $\varphi^* \leq \Phi_p$ where

$$(4.1.5)\quad \Phi_p = \varphi_p + \min\left[\alpha_x\alpha_y \max\left\{k_{IJ} - \frac{1}{\alpha_x} - \frac{1}{\alpha_y}, k_{I\bar{J}} - \frac{1}{\alpha_y}, k_{\bar{I}J} - \frac{1}{\alpha_x}, k_{\bar{I}\bar{J}}, 0\right\}\right.$$

$$\left., \max\left\{\frac{k_{IJ}}{4}\alpha^2 - \alpha, \frac{k_{\bar{I}\bar{J}}}{4}\alpha^2, \frac{k}{4}(\alpha - \tfrac{1}{k})^2\right\}\right]$$

where the quantities in this expression are defined as in (4.1.2) and (3.2.9).

Proof. By definition

$$\varphi^* = \varphi_p + \max\{\varphi(x,y) \mid x \in X_p, y \in Y_p\}$$

Since $\hat{g}^t x \leq \alpha_x$, $\forall\, x \in X_p$ and $\hat{h}^t y \leq \alpha_y$, $\forall\, y \in Y_p$,

$$\max\{\varphi(x,y) \mid x \in X_p, y \in Y_p\}$$

$$\leq \Omega_1 \equiv \max\{\varphi(x,y) \mid \hat{g}^t x \leq \alpha_x,\ x \geq 0,\ \hat{h}^t y \leq \alpha_y,\ y \geq 0\}$$

The problem defining $\Omega$ has the same structure as (3.3.12) and by



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Theorem 3.3.6, we have

$$\Omega_1 = \max [0, \max_{i,j} \{\bar{c}_i \alpha_x + \bar{d}_j \alpha_y + \bar{c}_{ij} \alpha_x \alpha_y\}]$$

$$= \max [0, \max \{\bar{c}_{ij} \alpha_x \alpha_y - \alpha_x - \alpha_y | (i,j) \in (I \times J) \cap K\} ,$$

$$\max \{\bar{c}_{ij} \alpha_x \alpha_y - \alpha_x | (i,j) \in (I \times \bar{J}) \cap K\} ,$$

$$\max \{\bar{c}_{ij} \alpha_x \alpha_y - \alpha_y | (i,j) \in (\bar{I} \times J) \cap K\} ,$$

$$\max \{\bar{c}_{ij} \alpha_x \alpha_y | (i,j) \in (\bar{I} \times \bar{J}) \cap K\}]$$

$$= \alpha_x \alpha_y \max \left\{ k_{IJ} - \frac{1}{\alpha_x} - \frac{1}{\alpha_y} , k_{I\bar{J}} - \frac{1}{\alpha_y} , k_{\bar{I}J} - \frac{1}{\alpha_x} , k_{\bar{I}\bar{J}} , 0 \right\} .$$

So

$$\varphi^* \leq \varphi_p + \Omega_1 \equiv \varphi_p + \alpha_x \alpha_y \max \left\{ k_{IJ} - \frac{1}{\alpha_x} - \frac{1}{\alpha_y} , k_{I\bar{J}} - \frac{1}{\alpha_y} , k_{\bar{I}J} - \frac{1}{\alpha_x} , k_{\bar{I}\bar{J}} , 0 \right\}$$

Similarly

$$\max \{\varphi(x,y) | \ x \in X_p , \ y \in Y_p\}$$

$$\leq \Omega_2 \equiv \max \{\varphi(x,y) | \ \hat{g}^t x + \hat{h}^t y \leq \alpha, \ x \geq 0, \ y \geq 0\}$$

since $\hat{g}^t x + \hat{h}^t y \leq \alpha$ for $\forall (x,y) \in X_p \times Y_p$ . By Theorem 3.2.3, we have

$$\Omega_2 = \max \left[ 0, \frac{k_{IJ}}{4} \alpha^2 - \alpha , \frac{k_{\bar{I}\bar{J}}}{4} \alpha^2 , \frac{k}{4} (\alpha - \frac{1}{k})^2 \right] .$$

So

$$\varphi^* \leq \varphi_p + \Omega_2 = \varphi_p + \max \left[ 0, \frac{k_{IJ}}{4} \alpha^2 - \alpha, \frac{k_{\bar{I}\bar{J}}}{4} \alpha^2, \frac{k}{4} (\alpha - \frac{1}{k})^2 \right]$$

and the result has been established. Q.E.D.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Corollary 4.1.2.1. If $c < 0$, $d < 0$ and if $g = \hat{g}$, $h = \hat{h}$, then $\varphi^* \leq \Phi_p$ where

$$\Phi_p = \varphi_p + \max\left[0, \min\left\{k_{IJ}s_x s_y - \frac{1}{k_{IJ}}, s\left(1 + \frac{k_{IJ}}{4}s\right)\right\}\right]$$

and

$$s_x = \alpha_x - \frac{1}{k_{IJ}}, \quad s_y = \alpha_y - \frac{1}{k_{IJ}}, \quad s = \alpha - \frac{4}{k_{IJ}}$$

Proof. In this case $\bar{I} = \emptyset = \bar{J}$ since $c < 0$ & $d < 0$. By Theorem 4.1.2,

$$\varphi^* \leq \varphi_p + \min\left[\max\left\{0, k_{IJ}\alpha_x\alpha_y - \alpha_x - \alpha_y\right\}, \max\left\{0, \frac{k_{IJ}}{4}\alpha^2 - \alpha\right\}\right]$$

$$= \varphi_p + \max\left[0, \min\left\{k_{IJ}\alpha_x\alpha_y - \alpha_x - \alpha_y, \frac{k_{IJ}}{4}\alpha^2 - \alpha\right\}\right]$$

Simple substitution of the variables gives the expression for $\Phi_p$.

Q.E.D.

The quantities in Theorem 4.1.2 and its corollary are obtained without any additional computation since $\alpha_x, \alpha_y$ and $\alpha$ have already been calculated to test the feasibility of the system (4.1.1). Note that if $\varphi_o - \Phi_p \leq \epsilon$ for some $\epsilon \geq 0$, then $\varphi_o$ is obviously an $\epsilon$-optimal solution.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

## 4.2 Finite Convergence of the Algorithm to an Extreme $\epsilon$-Optimal Solution

Based upon the results established thus far, let us state the algorithm in a compact fashion. Here the superscript $p$ refers to the $p^{th}$ locally maximum vertex.

Bilinear Programming Algorithm

Step 1. (Initialization).

Set $\varphi_o = \varphi_o' = \bar{\varphi} = -\infty$, $p=j=1$. Define a tolerance level $\epsilon > 0$ and a rounding level $\delta \geq 0$. Determine $x^{op} \in \text{ext } X_o$ and $y^{op} \in \text{ext } Y_o$ using the Phase I procedure to obtain a basic feasible solution of a linear program. If either $X_o$ or $Y_o$ is empty, then stop [(1.3) is infeasible]; otherwise go to Step 2.

Step 2. (Suboptimization).

Compute $x^{j+1,p}$, $y^{j+1,p}$ by (4.1.4) until either (i) or (ii) holds.

(i) there exists a ray incident to $(x^{k,p},y^{k,p})$ in whose direction $\varphi(x,y)$ can increase without bound;

(ii) $(x^{k+1,p},y^{k+1,p}) = (x^{k,p},y^{k,p})$.

If (i) obtains, then stop [(1.3) has an unbounded solution]. If (ii) obtains, let $(x^{\infty,p},y^{\infty,p}) = (x^{k,p},y^{k,p})$ and go to Step 3.

Step 3. (Canonical Representation).

Generate a canonical form (2.1.4) relative to $(x^{\infty,p},y^{\infty,p})$. If $C \leq 0$, then stop [$(x^{\infty,p},y^{\infty,p})$ is an optimal solution]. Otherwise go to Step 4.

Step 4. (Test of the Local Maximality).

Calculate $\varphi^p_{i_o j_o}$ defined by (2.2.6).



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

4.1: If $\varphi^p_{i_o j_o} > 0$ .

Increase $x_{i_o}$ and $y_{j_o}$ up to $\bar{x}_{i_o}$ and $\bar{y}_{j_o}$, respectively, where $\bar{x}_{i_o}$ and $\bar{y}_{j_o}$ are defined by (2.2.5), i.e., move to the pair of adjacent vertices $x^{op}$ and $y^{op}$ .

Go back to Step 2.

4.2: If $\varphi^p_{i_o j_o} \leq 0$ [$(x^{\infty p}, y^{\infty p})$ is a locally maximum vertex]

Define $(x^p, y^p) = (x^{\infty p}, y^{\infty p})$ , $\varphi_p = \varphi(x^p, y^p)$ and go to Step 5.

Step 5. (Construction of the Cutting Planes)

Update $\varphi_o : \varphi_o = \max \{\varphi_o, \varphi_p\}$ . Define $\varphi_o' = \varphi_o + \epsilon$ and construct the new constraints $(g^p)^t x \geq \sigma_x^*$ , $(h^p)^t y \geq \sigma_y^*$ , $(g^p)^t x + (h^p)^t y \geq \sigma^*$ by the method described in Sections 3.2 and 3.3 by using $\varphi_o'$ instead of $\varphi_o$ . Go to Step 6.

Rules for the choice of the vectors $g^p$ and $h^p$

Rule 1. Use $g^p = g^{p,\delta}$ and $h^p = h^{p,\delta}$ for all $p$ where

$$g_i^{p,\delta} = \begin{cases} \dfrac{c_i^p}{c_o^p} & \text{if } \dfrac{c_i^p}{c_o^p} \geq \delta \\ \delta & \text{if } \dfrac{c_i^p}{c_o^p} < \delta \end{cases}$$

$$h_i^{p,\delta} = \begin{cases} \dfrac{d_i^p}{d_o^p} & \text{if } \dfrac{d_i^p}{d_o^p} \geq \delta \\ \delta & \text{if } \dfrac{d_i^p}{d_o^p} < \delta \end{cases}$$

where $\delta$ is the rounding level and $c_o^p = \min_{1 \leq i \leq m} c_i^p$ , $d_o^p = \min_{1 \leq j \leq n} d_i^p$ .



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Rule 2. If $x^p \notin \text{ext } X_o$ then use $g^p = \bar{g}^p$ by ignoring all but one of the adjoined constraints which are tight at $x^p$. Similarly, if $y^p \notin \text{ext } X_o$, then use $h^p = \bar{h}^p$ by ignoring all but one of the adjoined constraints which are tight at $y^p$. Here $\bar{g}^p$ and $\bar{h}^p$ are the normalized (i.e., $\max_{1\le i\le m} \bar{g}_i^p = \max_{1\le j\le n} \bar{h}_j^p = 1$) vectors proportional to the ones defined by (3.3.9) and (3.3.10).

Step 6. (Optimality Test and the Computation of an Upper Bound)

Compute $\alpha_x^p, \alpha_y^p, \alpha^p$ and $x^{o,p+1}, y^{o,p+1}$ defined by (4.1.2).

6.1: $(\alpha_x^p, \alpha_y^p, \alpha^p) \not\ge (\sigma_x^{*p}, \sigma_y^{*p}, \sigma^{*p})$. [$\varphi_o$ and the corresponding pair of vectors $(x^*, y^*)$ give an $\epsilon$-optimal solution]. Go to Step 7.

6.2: $(\alpha_x^p, \alpha_y^p, \alpha^p) \ge (\sigma_x^{*p}, \sigma_y^{*p}, \sigma^{*p})$.

Compute $\Phi_p$ by (4.1.5) and update the upper bound $\bar{\varphi} := \min [\bar{\varphi}, \Phi_p]$

6.2.1: If $\bar{\varphi} \le \varphi_o + \epsilon$ [$\varphi_o$ and the corresponding vectors $(x^*, y^*)$ give an $\epsilon$-optimal solution]. Go to Step 7.

6.2.2: If $\bar{\varphi} > \varphi_o + \epsilon$.

Let $X_{p+1} = \{x \mid x \in X_p, (g^p)^t x \ge \sigma_x^{*p}\}$, $Y_{p+1} = \{y \mid y \in Y_p, (h^p)^t y \ge \sigma_y^{*p}\}$ and go to Step 2.

Step 7. (Construction of an Extreme $\epsilon$-Optimal Solution)

Compute $(\bar{x}^o, \bar{y}^o)$, an extreme $\epsilon$-optimal solution, by solving a pair of linear programs: maximize $\{\varphi(x, y^o) \mid x \in X_o\}$, maximize $\{\varphi(\bar{x}^o, y) \mid y \in Y_o\}$ and stop. (Obviously if $\varphi(\bar{x}^o, \bar{y}^o) \ge \varphi(x^o, y^o) + \epsilon$, then $(\bar{x}^o, \bar{y}^o)$ is an optimal solution of (1.3).)



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Theorem 4.2.1. The algorithm defined above terminates in finitely many steps if $X_o$ and $Y_o$ are bounded.

Proof. Obviously, if either of the constraints $(g^p)^t x \geq \sigma_x^{*p}$ or $(h^p)^t y \geq \sigma_y^{*p}$ to be adjoined at the $p^{th}$ locally maximum vertex $(x^p, y^p)$ eliminates certain finite regions of $X_p$ or $Y_p$ containing $x^p$ or $y^p$, then either $X_p$ or $Y_p$ (or both) will eventually shrink to an empty set (Step 6.1 obtains) and the algorithm will terminate. Since $g^p$ and $h^p$ have been normalized ($\|g^p\|_\infty = \|h^p\|_\infty = 1$) for all $p$, the finite termination will be guaranteed if $\min\{\sigma_x^{*p}, \sigma_y^{*p}\}$ is bounded strongly away from zero for all $p$. We proved the following results in Theorems 3.3.4 and 3.3.4': if $\varphi_o - \varphi_p \geq \epsilon$, then $\sigma_x^{*p} \geq \epsilon/\alpha_p K_x^p$ and $\sigma_y^{*p} \geq \epsilon/\beta_p K_y^p$ where $K_x^p = \max\{|c_{ij}^p/g_i^p| \; i=1,\ldots,m;\; j=1,\ldots,n\}$,

$K_y^p = \max\{|c_{ij}^p/h_j^p| \; i=1,\ldots,m;\; j=1,\ldots,n\}$, $\alpha_p = \max\left\{\sum_{j=1}^{n} y_j \,|\, y \in Y_p\right\}$

and

$$\beta_p = \max\left\{\sum_{i=1}^{m} x_i \,|\, x \in X_p\right\}.$$

We used $\varphi' = \varphi_o + \epsilon$ instead of $\varphi_o$ to determine the constants $\sigma_x^{*p}$ and $\sigma_y^{*p}$ for all $p$, so that the results of Theorems 3.3.4, 3.3.4' apply here. Hence the finite termination will be established if we show the boundedness of $\alpha_p K_x^p$ or $\beta_p K_y^p$ for all $p$. By assumption, $X_o$ and $Y_o$ are bounded and so are $X_p \subset X_o$, $Y_p \subset Y_o$, so that $\alpha_p$ and $\beta_p$ are always bounded above by a constant. Thus it suffices to show the boundedness of $K_x^p$ or $K_y^p$. Let us prove it for $K_x^p$.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

(i) Boundedness of $c_{ij}^p$ for all i,j and p . Assume $c_{ij}^p \to \infty$ for some i,j . By the expression following (2.1.3), this can happen if and only if the basis defining $x^p$ or $y^p$ tends to a singular matrix. Geometrically, this happens if, and only if any two of the hyperplanes defining $x^p$ or $y^p$ tends to be parallel to each other. Figure 4.2.1 illustrates this kind of situation. We will show that this cannot happen in our algorithm. If $x^p \in \text{ext } X_o$ and $y^p \in \text{ext } Y_o$ , then by definition the bases defining $x^p$ and $y^p$ are nonsingular and $c_{ij}^p$ is bounded above for all i,j . On the other hand, if $x^p \notin \text{ext } X_o$ , then a cut eliminating $x^p$ and passing through the m adjacent vertices will be introduced since we can assume without loss of generality (by temporarily ignoring all but one of the adjoined constraints which are tight at $x^p$ , if any) that $x^p$ is a non-degenerate vertex (Fig. 4.2.2).

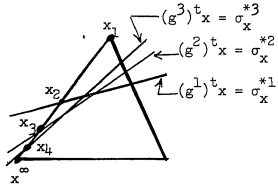



Figure 4.2.1

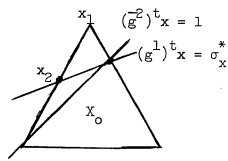



Figure 4.2.2



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Hence a situation like Figure 4.2.1 will be avoided and the basis defining $x^p$ will never approach to a singular matrix.

(ii) Boundedness of $1/g_i^p$ for all $i$ and $p$. For the choice of $g^p = \hat{g}^{p,\delta}$, $g_i^p \geq \delta$ for all $i$ and $p$ by definition. So, for this particular choice of $g^p$, the boundedness of $1/g_i^p$ is guaranteed. (Note that we adjoin this type of cut for all $p$.)

Boundedness of $c_{ij}^p$ for all $i,j$ and $p$ and boundedness of $1/g_i^p$ for particular choice of $g^p = \hat{g}^{p,\delta}$ guarantees the boundedness of $K_x^p$ for all $p$. Q.E.D.

Theorem 4.2.2. The algorithm defined above generates an extreme $\epsilon$-optimal solution in finitely many steps.

Proof. By Theorem 4.2.1 the algorithm terminates in finitely many steps. When Step 7 is reached via Step 6.2.1, then the $\epsilon$-optimality of $(x^*,y^*)$ follows from Theorem 4.1.2. If, on the other hand, Step 7 is reached via Step 6.1, then by Theorem 4.1.1, there exists no pair of vectors $(x,y) \in X_o \times Y_o$ such that $\varphi(x,y) \geq \varphi_o' = \varphi_o + \epsilon$, so that $\varphi_o$ and the corresponding pair of vectors $(x^*,y^*)$ are $\epsilon$-optimal. In either case, when we arrive at Step 7, we have an $\epsilon$-optimal solution. That Step 7 generates an extreme $\epsilon$-optimal solution is obvious from the proof of Theorem 2.1.1. Q.E.D.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

# 5. EXTENSIONS OF BILINEAR PROGRAMMING

## 5.1 Extended Bilinear Programming Problem

### 5.1.1 Definitions and Basic Properties of the Extended Bilinear Programming Problem

As an extension of the bilinear programming problem, let us consider the following extended bilinear programming problem (EBLP):

$$(5.1.1)\qquad \begin{cases} \text{maximize} \quad \varphi(x^1,\ldots,x^N) = \sum_{i=1}^{N} (c^i)^t x^i + \sum_{i=1}^{N}\sum_{j=i+1}^{N} (x^i)^t C^{ij} x^j \\ \text{subject to} \quad A_i x^i \le b^i \ , \quad x^i \ge 0 \ , \quad i=1,\ldots,N \end{cases}$$

where $c^i,\ x^i \in R^{n_i}$, $C^{ij} \in R^{n_i \times n_j}$, $A_i \in R^{k_i \times n_i}$. Sometimes we will call (5.1.1) the extended bilinear programming problem of order $N$. When we put this into the form of the general quadratic programming problem (1.5), matrices $Q$ and $A$ have the following special structure

$$(5.1.2)\quad Q = \begin{bmatrix} 0 & C^{12} & C^{13} & \cdots & C^{1,N} \\ (C^{12})^t & 0 & C^{23} & \cdots & C^{2,N} \\ (C^{13})^t & (C^{23})^t & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 & C^{N-1,N} \\ (C^{1N})^t & \cdots & (C^{N-1,N})^t & 0 & \end{bmatrix}, \quad A = \begin{bmatrix} A_1 & & & 0 \\ & A_2 & & \\ & & \ddots & \\ 0 & & & A_N \end{bmatrix}$$

Let

$$(5.1.3)\qquad X_i = \{x^i \in R^{n_i} \mid A_i x^i \le b^i \ , \quad x^i \ge 0\} \ , \quad i=1,\ldots,N \ .$$

As in the BLP case, the following theorems hold.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Theorem 5.1.1. If EBLP (5.1.1) has a finite optimal solution, then there exists an optimal solution $x_* = (x_*^1, \ldots, x_*^N)$ such that $x_*^i \in \text{ext } X_i$ .

Theorem 5.1.2. If $b^i \geq 0$ , i=1,...,N, then the origin $(0_{n_1}, \ldots, 0_{R_n})$ of the system (5.1.1) is

(i) a Kuhn-Tucker point if $c^i \leq 0$ , i=1,...,N ;

(ii) a local maximum point if (a), (b) hold

(a) $c^i \leq 0$ , i=1,...,N

(b) either $c_p^i < 0$ or $c_q^j < 0$ if $(p,q)^{th}$ element of $C^{ij}$ is positive.

(iii) a global maximum if $C^{ij} \leq 0$ for all i and j .

The proofs of these theorems are analogous to those of Theorems 2.1.1 and 2.1.2 and will be omitted here. The following theorem is the stronger version of the statement (iii) of Theorem 5.1.2.

Theorem 5.1.3. Assume $b^i \geq 0$ . If $C^{ij} \leq 0$ for j=1+1,...,N and $C^{ki} \leq 0$ for k=1,...,i-1, then there exists an optimal solution $x_* = (x_*^1, \ldots, x_*^N)$ such that $x_*^i = 0$ .

Proof.

$$\varphi(x^1, \ldots, x^i, \ldots, x^N) = (x^i)^t \left\{ \sum_{j=i+1}^{N} C^{ij} x^j + \sum_{k=1}^{i-1} (C^{ki})^t x^k \right\} + \varphi(x^1, \ldots, x^{i-1}, 0, x^{i+1}, \ldots, x^N)$$

$$\leq \varphi(x^1, \ldots, x^{i-1}, 0_{n_i}, x^{i+1}, \ldots, x^N)$$

for all $x^j \geq 0$ , $j \neq i$ . Hence

$$\max \{\varphi(x^1, \ldots, x^i, \ldots, x^N) \mid x^j \in X_j,\ j=1,\ldots,N\}$$

$$\leq \max \{\varphi(x^1, \ldots, x^{i-1}, 0_{n_i}, x^{i+1}, \ldots, x^N) \mid x^j \in X_j,\ j=1,\ldots,N,\ j \neq i\}$$

Q.E.D.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Let us define the notion of the locally maximum vertex analogously as before.

Definition 5.1.1. The point $\tilde{x} = (\tilde{x}^1, \ldots, \tilde{x}^N)$ where $\tilde{x}^i \in \text{ext } X_i$, $i=1,\ldots,N$ is called a locally maximum vertex if

(i) $c_i \leq 0$ , $i=1,\ldots,N$

(ii) $\varphi(\tilde{x}) \geq \varphi(\tilde{x}^1, \ldots, \tilde{x}^{i-1}, x^i, \tilde{x}^{i+1}, \ldots, \tilde{x}^{j-1}, x^j, \tilde{x}^{j+1}, \ldots, \tilde{x}^N)$ holds for all $x^i \in N_{X_i}(\tilde{x}_i) \cap R_{X_i}(\tilde{x}^i)$ and $x^j \in N_{X_j}(\tilde{x}_j) \cap R_{X_j}(\tilde{x}_j)$ for any pair of indices $i$ and $j$ .

The canonical form relative to the point $x = (x^1, \ldots, x^N)$ where $x^i \in \text{ext } X_i$ will also be defined as before.

5.1.2 The Cutting Plane Algorithm for the Extended Bilinear Programming Problem

Let us consider the extension of the BLP algorithm to the EBLP (5.1.1). As will be obvious from its combinatorial nature, the difficulty of solving this problem increases as its order $N$ increases. So we will discuss the case $N = 3$ in somewhat detail and leave the higher order cases as an obvious (but not trivial) extension of the case $N = 3$ .

a. Construction of a Cutting Plane

Assume that we obtained a locally maximum vertex and that we have a canonical form relative to this point



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$
(5.1.4)\quad
\begin{cases}
\text{maximize} \quad \varphi(x^1,x^2,x^3) \\
\qquad = (c^1)^t x^1+(c^2)^t x^2+(c^3)^t x^3+(x^1)^t C^{12} x^2+(x^1)^t C^{13} x^3+(x^2)^t C^{23} x^3 \\
\text{subject to} \qquad A_1 x^1 \leq b^1 \ , \ x^1 \geq 0 \ , \\
\qquad\qquad A_2 x^2 \leq b^2 \ , \ x^2 \geq 0 \ , \\
\qquad\qquad A_3 x^3 \leq b^3 \ , \ x^3 \geq 0 \ , \\
\qquad (c^i \leq 0 \ , \ b^i \geq 0 \ , \ i=1,2,3.)
\end{cases}
$$

As before, we try to construct a cutting plane which eliminates the origin and yet does not eliminate any feasible point which may possibly generate a better objective value.

Let $g^i > 0$ , i=1,2.3 and define $\psi(\sigma)$ by

$$
(5.1.5)\quad \psi(\sigma) = \max\{\varphi(x^1,x^2,x^3) \mid (g^1)^t x^1+(g^2)^t x^2+(g^3)^t x^3 \leq \sigma \ , \ x^i \geq 0, \ i=1,2,3\}
$$

i.e., the maximum objective value of the parametric EBLP:

$$
(5.1.6)\quad
\begin{cases}
\text{maximize} \quad \varphi(x^1,x^2,x^3) \\
\qquad = (c^1)^t x^1+(c^2)^t x_2+(c^3)^t x^3+(x^1)^t C^{12} x^2+(x^1)^t C^{13} x^3+(x^2)^t C^{23} x^3 \\
\text{subject to} \qquad (g^1)^t x^1 + (g^2)^t x^2 + (g^3)^t x^3 \leq \sigma \\
\qquad\qquad x^1 \geq 0 \ , \ x^2 \geq 0 \ , \ x^3 \geq 0
\end{cases}
$$

Let

$$
(5.1.7)\qquad \sigma^* = \max\{\sigma' \mid \psi(\sigma) \leq \varphi_o - \varphi_p \ , \ \sigma \in [0,\sigma']\}
$$

then the cut

$$
(5.1.8)\qquad (g^1)^t x^1 + (g^2)^t x^2 + (g^3)^t x^3 \geq \sigma^*
$$

is legitimate if $\sigma^* > 0$ .



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Theorem 5.1.4. The global maximum of (5.1.6) is attained either at the origin or else there exists an optimal solution $x_* = (x_*^1, x_*^2, x_*^3)$ such that

(i) $(g^1)^t x_*^1 + (b^2)^t x_*^2 + (g^3)^t x_*^3 = \sigma$;

(ii) $x_*^i$ contains at most one positive component for $i=1,2,3$.

The proof is along the same line as that of Theorem 3.2.1 and will be omitted here. By virtue of this theorem, it is enough to consider the set of vectors $x^1, x^2, x^3$ of the form $x^i = (0, \ldots, 0\ x^i_{k_i}, 0, \ldots, 0)$, $i=1,2,3$, to calculate $\psi(\sigma)$. Thus it suffices to solve the following easier problem:

$$
(5.1.9)\quad
\begin{cases}
\underset{k_1,k_2,k_3}{\text{maximize}} \left\{ \begin{array}{l} \text{maximize } \tilde{\Phi}(x^1_{k_1}, x^2_{k_2}, x^3_{k_3}) \\ \qquad = c^1_{k_1} x^1_{k_1} + c^2_{k_2} x^2_{k_2} + c^3_{k_3} x^3_{k_3} + c^{12}_{k_1 k_2} x^1_{k_1} x^2_{k_2} + c^{13}_{k_1 k_3} x^1_{k_1} x^3_{k_3} + c^{23}_{k_2 k_3} x^2_{k_2} x^3_{k_3} \end{array} \right\} \\
\text{subject to} \quad g^1_{k_1} x^1_{k_1} + g^2_{k_2} x^2_{k_2} + g^3_{k_3} x^3_{k_3} = \sigma \\
\qquad\qquad x^1_{k_1} \geq 0\ ,\ x^2_{k_2} \geq 0\ ,\ x^3_{k_3} \geq 0
\end{cases}
$$

which is equivalent to

$$
(5.1.10)\quad
\begin{cases}
\underset{j,k,\ell}{\text{maximize}} \left\{ \begin{array}{l} \text{maximize } \tilde{\tilde{\Phi}}(y^1_j, y^2_k, y^3_\ell) \\ \qquad = \bar{c}^1_j y^1_j + \bar{c}^2_k y^2_k + \bar{c}^3_\ell y^3_\ell + \bar{c}^{12}_{jk} y^1_j y^2_k + \bar{c}^{13}_{j\ell} y^1_j y^3_\ell + \bar{c}^{23}_{k\ell} y^2_k y^3_\ell \end{array} \right\} \\
\text{subject to} \quad y^1_j + y^2_k + y^3_\ell = \sigma\ ,\ y^1_j \geq 0\ ,\ y^2_k \geq 0\ ,\ y^3_\ell \geq 0
\end{cases}
$$

where $\bar{c}^1_j = c^1_j / g^1_j$, $\bar{c}^{12}_{jk} = c^{12}_{jk} / g^1_j g^2_k$, etc. Let us assume that $c^i < 0$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

for all $i$ and let $g^i = -c^i$, $i=1,2,3$. Then $c^1_j = c^2_k = c^3_\ell = -1$ and

$$\psi(\sigma) = \max_{j,k,\ell} \left\{ \max_{y^1_j, y^2_k, y^3_\ell} \tilde{\varphi}(y^1_j, y^2_k, y^3_\ell) = -\sigma + c^{12}_{jk} y^1_j y^2_k + c^{13}_{j\ell} y^1_j y^3_\ell + c^{23}_{k\ell} y^2_k y^3_\ell \;\middle|\; y^1_j + y^2_k + y^3_\ell = \sigma,\ y^1_j \geq 0,\ y^2_k \geq 0,\ y^3_\ell \geq 0 \right\} \tag{5.1.11}$$

Let

$$\zeta(\sigma) = \max \{ f(y_1, y_2, y_3) = c_3 y_1 y_2 + c_2 y_1 y_3 + c_1 y_2 y_3 \mid y_1 + y_2 + y_3 = \sigma,\ y_1 \geq 0,\ y_2 \geq 0,\ y_2 \geq 0 \} \tag{5.1.12}$$

where

$$c_3 = \max \{ c^{12}_{jk} \mid j=1, \ldots, n_1,\ k=1, \ldots, n_2 \} \tag{5.1.13}$$

etc. Then $\psi(\sigma) \leq \zeta(\sigma) - \sigma$ for all $\sigma \geq 0$ since $\tilde{\varphi}$ is larger for larger $c^{12}_{jk}$, $c^{13}_{k\ell}$, and $c^{23}_{k\ell}$ for fixed $y^1_j$, $y^2_k$ and $y^3_\ell$. Let us consider the right hand side of the definition of $\zeta(\sigma)$ in more detail:

$$\begin{cases} \text{maximize} & f(y_1, y_2, y_3) = c_1 y_2 y_3 + c_2 y_1 y_3 + c_3 y_1 y_2 \\ \text{subject to} & y_1 + y_2 + y_3 = \sigma,\ y_1 \geq 0,\ y_2 \geq 0,\ y_3 \geq 0 \end{cases} \tag{5.1.14}$$

Let

$$Y(\sigma) = \{ (y_1, y_2, y_3) \mid y_1 + y_2 + y_3 = \sigma,\ y_1 \geq 0,\ y_2 \geq 0,\ y_3 \geq 0 \} \tag{5.1.15}$$

We can assume without loss of generality that at most one of $c_1, c_2, c_3$ are non-positive since if $c_1 \leq 0$ and $c_2 \leq 0$, then $c^{23} \leq 0$, $c^{13} \leq 0$ in (5.1.4) and by Theorem 5.1.3 there exists an optimal solution $(x^1_*, x^2_*, 0_{n_3})$ of (5.1.4) so that the problem is reducible to a bilinear programming problem. Let us assume without loss of generality that $c_1 \leq c_2 \leq c_3$ in the following.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Theorem 5.1.5. $\zeta(\sigma)$ is given by

$$\zeta(\sigma) = \begin{cases} \dfrac{c_1c_2c_3}{2(c_2c_3 + c_3c_1 + c_1c_2) - (c_1^2 + c_2^2 + c_3^2)}\,\sigma^2 & \text{if } c_3 \geq c_2 \geq c_1 \geq 0 \\ & \quad \& \; c_3 \geq c_2 + c_1 \\ \dfrac{c_3}{4}\,\sigma^2 & \text{otherwise} \end{cases}$$

(5.1.16)

Proof. Case 1. $c_1 < 0 \leq c_2 \leq c_3$ . For all $y = (y_1,y_2,y_3) \in Y(\sigma)$ , we have $f(y_1,y_2,y_3) = c_1y_2y_3+c_2y_3y_1+c_3y_1y_2 \leq c_2y_3y_1+c_3y_1y_2 \leq c_3(y_1y_3+y_1y_2)$ . Hence

$$\zeta(\sigma) = \max\{f(y) \mid y \in Y(\sigma)\} \leq \max\{c_3(y_1y_3+y_1y_2) \mid y \in Y(\sigma)\}$$
$$= \max\{c_3y_1(\sigma-y_1) \mid 0 \leq y_1 \leq \sigma\} = \frac{c_3}{4}\,\sigma^2$$

But $f(1/2\,\sigma,\, 1/2\,\sigma,0) = \dfrac{c_3}{4}\,\sigma^2$ , so $\zeta(\sigma) = \dfrac{c_3}{4}\,\sigma^2$.

Case 2. $0 \leq c_1 \leq c_2 \leq c_3$ . Let $L(y,\lambda)$ be the Lagrangean function associated with (5.1.14).

$$L(y;\lambda) = c_1y_2y_3 + c_2y_3y_1 + c_3y_1y_2 + \lambda(1 - y_1 - y_2 - y_3)$$

We solve the system $\partial L/\partial y_i = 0$ , i=1,2,3 to obtain

$$y_1^* = k\,c_1(c_2 + c_3 - c_1)\sigma$$
$$y_2^* = k\,c_2(c_3 + c_1 - c_2)\sigma$$
$$y_3^* = k\,c_3(c_1 + c_2 - c_3)\sigma$$

where

$$k = \frac{1}{2(c_2c_3 + c_3c_1 + c_1c_2) - (c_1^2 + c_2^2 + c_3^2)} \quad .$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Case 2.1. $c_1 + c_2 \le c_3$ , $c_3 \ge c_2 \ge c_1 \ge 0$

In this case, we have $y_1^* \ge 0$ , $y_2^* \ge 0$ , $y_3^* \ge 0$ and

$$\zeta(\sigma) = f(y_1^*, y_2^*, y_3^*) = k\, c_1 c_2 c_3\, \sigma^2 = \frac{c_1 c_2 c_3}{2(c_2 c_3 + c_3 c_1 + c_1 c_2) - (c_1^2 + c_2^2 + c_3^2)}\, \sigma^2$$

Case 2.2. $c_3 \ge c_2 + c_1$ , $c_3 \ge c_2 \ge c_1 \ge 0$ . For all $y \in Y$ , we have

$$\begin{aligned} f(y_1, y_2, y_3) &= c_1 y_2 y_3 + c_2 y_3 y_1 + c_3 y_1 y_2 \le c_2 y_3 (y_1 + y_2) + c_3 y_1 y_2 \\ &= -c_2 \left( y_1 - \frac{c_3 - 2c_2}{2c_2}\, y_2 \right)^2 + \frac{c_3 (c_3 - 4c_2)}{4c_2}\, y_2^2 + c_2 \end{aligned}$$

Hence

$$\begin{aligned} \zeta(\sigma) &\le \max \left\{ -c_2 \left( y_1 - \frac{c_3 - 2c_2}{2c_2}\, y_2 \right)^2 + \frac{c_3 (c_3 - 4c_2)}{4c_2}\, y_2^2 + c_2 \,\middle|\, y_1 + y_2 \le \sigma , \right. \\ &\qquad \left. y_1 \ge 0 , \; y_2 \ge 0 \right\} \\ &\le \max \left\{ \frac{c_3 (c_3 - 4c_2)}{4c_2}\, y_2^2 + c_2 \,\middle|\, y_1 = \frac{c_3 - 2c_2}{2c_2}\, y_2 , \; y_1 + y_2 \le \sigma , \; y_1 \ge 0 , \; y_2 \ge 0 \right\} \\ &= 2 \left( c_2 - \frac{2c_2^2}{c_3} \right) \sigma^2 \le \frac{c_3}{4}\, \sigma^2 \end{aligned}$$

But $f(1/2\,\sigma, 1/2\,\sigma, 0) = \frac{c_3}{4}\, \sigma^2$ , so $\zeta(\sigma) = \frac{c_3}{4}\, \sigma^2$ . Q.E.D.

Let

$$\tau' = \max \{ \sigma \,|\, \zeta(\sigma) - \sigma \le \varphi_o - \varphi_p , \; \sigma \in [0, \tau'] \}$$

then the cut

$$-(c^1)^t x^1 - (c^2)^t x^2 - (c^3)^t x^3 \ge \tau'$$

is legitimate since $\psi(\sigma) \le \zeta(\sigma) - \sigma$ for all $\sigma \ge 0$ so that $\tau' \le \tau$ and $\tau' > 0$ by (5.1.16). The expression of $\zeta(\sigma)$ for the degenerate situation is complicated and will not be given here.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The cutting plane generated above necessarily destroys the separable structure of the constraint set and cutting planes of the form $(g^i)^t x^i \geq \tau_{x^i}$, i=1,2,3 are highly desirable. Theoretically, a 'deep' cut of this form can be obtained if we can solve a bilinear programming problem with parametric objective functions, but this seems to require a prohibitive amount of computation and it may be impractical from the computational point of view unless the size of the problem is very small or an efficient algorithm for solving a parametric bilinear programming becomes available.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

b. Extended Bilinear Programming Algorithm

One version of the extended bilinear programming algorithm may be stated as follows:

Step 1. (Initialization). Set $\varphi_o = -\infty$ , $p = j = 1$ . Determine $x^i_{op} \in \text{ext } X_i$ using the Phase I procedure to obtain a basic feasible solution of a linear program. If either one of $X_i$ , $i=1,\ldots,N$ is empty, then stop [(5.1.1) is infeasible]; otherwise go to Step 2.

Step 2. (Suboptimization). Compute $x^i_{j+1,p}$ , $i=1,\ldots,N$ by

$$x^i_{j+1,p} = \arg\max \{\varphi(x^1_{j+1,p}\ldots,x^{i-1}_{j+1,p},x^i,x^{i+1}_{j,p},\ldots,x^N_{j,p}) | x^i \in X_i\}$$

until either of the following holds for some $k$ .

(i) there exists a ray incident to $(x^1_{k,p}\ldots,x^{i-1}_{k,p},x^i_{k-1,p},\ldots,x^N_{k-1,p})$ in whose direction $\varphi(x^1,\ldots,x^N)$ can increase without a bound.

(ii) $x^i_{k,p} = x^i_{k-1,p}$ for some $i$ .

If (i) obtains, then stop [(5.1.1) has an unbounded solution]. If (ii) obtains, let $(x^1_{\infty,p},\ldots,x^N_{\infty,p}) = (x^i_{k,p},\ldots,x^i_{k,p},x^{i+1}_{k-1,p},\ldots,x^N_{k-1,p})$ and go to Step 3.

Step 3. (Canonical Expression). Generate a canonical form relative to $(x^1_{\infty,p},\ldots,x^N_{\infty,p})$ . If there exists an index $i$ such that $c^{ij} \leq 0$ for $j=i+1,\ldots,N$ and $c^{ki} \leq 0$ for $k=1,\ldots,i-1$ , then fix $x^i = 0$ and reduce the problem to an EBLP of order $N-1$ (Theorem 5.1.3). Reduce the order as long as this is possible. In particular, if $c^{ij} \leq 0$ for all $i$ and $j$ , then stop [$(x^1_{\infty p},\ldots,x^N_{\infty,p})$ is an optimal solution]. Otherwise go to Step 4.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Step 4. (Test of the Local Maximality). Calculate

$$\varphi_{ij} \equiv c^i_{r_i}\bar{x}^i_{r_i} + c^j_{s_j}\bar{x}^j_{s_j} + C^{ij}_{r_i s_j}\bar{x}^i_{r_i}\bar{x}^j_{s_j}$$

$$= \max \{c^i_r\bar{x}^i_r + c^j_s\bar{x}^j_s + C^{ij}_{rs}\bar{x}^i_r\bar{x}^j_s \mid C^{ij}_{rs} > 0 \ ; \ r=1,\ldots,n_i \ ; \ s=1,\ldots,n_j\}$$

where $\bar{x}^i_r$ represents the coordinate of the vertex adjacent to the origin in the direction of $x^i_r$.

4.1: $\varphi_{ij} > 0$ for some $i$ and $j$.
Increase $x^i_{r_i}$ and $x^j_{r_j}$ up to $\bar{x}^i_{r_i}$ and $\bar{x}^j_{r_j}$, respectively and go back to Step 2.

4.2: $\varphi_{ij} \leq 0$ for all $i$ and $j$. $[(x^1_{\omega p},\ldots,x^N_{\omega p})$ is a locally maximum vertex]. Define $(x^1_p,\ldots,x^N_p) = (x^1_{\omega p},\ldots,x^N_{\omega,p})$, $\varphi_p = \varphi(x^1_p,\ldots,x^N_p)$ and go to Step 5.

Step 5. (Construction of a Cutting Plane).

Update $\varphi_o : \varphi_o = \max\{\varphi_o,\varphi_p\}$. Construct a cutting plane $\sum_{i=1}^N (g^i)^t x^i \geq \tau$ by the method developed in (a) of this section. Go to Step 6.

Step 6. (Optimality Test). Compute

$$\alpha \equiv \sum_{i=1}^N (g^i)^t x^{i,p+1} = \max\left\{\sum_{i=1}^N (g^i)^t x^i \,\middle|\, x^i \in X_i, \ i=1,\ldots,N: \ \sum_{i=1}^N G^i x^i \geq t\right\}$$

where $\sum_{i=1}^N G^i x^i \geq t$ is the constraints corresponding to the cuts adjoined at the previous $p-1$ locally maximum vertex.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

6.1: $\alpha < \tau$ . [$\varphi_o$ and the corresponding vector $(\bar{x}^1, \ldots, \bar{x}^N)$ give an optimal solution]. Go to Step 7.

6.2: $\alpha \geq \tau$ . Go to Step 2.

Step 7. (Construction of an Extreme Optimal Solution)

Compute $(\bar{x}^1, \ldots, \bar{x}^N)$ , an extreme optimal solution by solving N linear programs $\max \{\varphi(\bar{x}^1, \ldots, \bar{x}^{i-1}, x^i, x_{op}^{i+1}, \ldots, x_{op}^N) | x^i \in X_i\}$ , $i=1, \ldots, N$ and stop.

Unlike the BLP case, the finite convergence of this algorithm to an optimal solution or even to an $\epsilon$-optimal solution is not guaranteed and the algorithm may cycle on certain sets of locally maximum vertices, but it will sometimes generate and identify an optimal solution for a certain nice class of problems.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

## 5.2 Other Extensions

In this section we will briefly look into the extensions of the BLP algorithm to more general problems.

### 5.2.1 Extension to a Slightly More General Quadratic Programming Problem

Let us consider the QP :

$$(5.2.1) \qquad \begin{cases} \text{maximize} & \varphi(x,y) = c^t x + d^t y + \frac{1}{2} x^t P x + x^t C y \\ \text{subject to} & Ex \leq e \ , \quad x \geq 0 \\ & Fy \leq f \ , \quad y \geq 0 \end{cases}$$

The only difference between this and BLP (1.3) is the existence of the additional quadratic term $\frac{1}{2} x^t P x$ in $\varphi(x,y)$ . We will assume here that the $m \times m$ matrix $P$ is a symmetric positive semi-definite matrix.and that the constraint set is non-empty.

Theorem 5.2.1. If $X_o = \{x | Ex \leq e, x \geq 0\}$ and $Y_o = \{y | Fy \leq f, y \geq 0\}$ are bounded, then there exists an optimal solution $(x^*, y^*)$ of (5.2.1) such that $x^* \in \text{ext } X_o$ and $y^* \in \text{ext } Y_o$ .

Proof. Follows from the convexity of $f$ for fixed $x$ or fixed $y$ and from the existence of the finite optimal solution. Q.E.D.

Owing to this key property, we can extend the BLP algorithm to (5.2.1) in an almost straightforward manner. The major differences from the algorithm stated in Chapter 4 are as follows:

(i) The suboptimization process (Step 2) to compute $x^{i+1}$

To determine an improved solution $x^{i+1}$ by the formula analogous to (4.1.4), i.e.,



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

(5.2.2) $$x^{i+1} = \arg\max_{x} \{\varphi(x,y^i) \mid x \in X_p\}$$

we have to solve a maximization problem of a convex quadratic function over a polyhedral convex set (actually, this can be accomplished by the application of the BLP algorithm: see § 4.1 of Part II for details). But what we really need as the outcome of the suboptimization process is a locally maximum vertex yielding a better objective value rather than a global maximum of a difficult subproblem defined by (5.2.2). So we will replace (5.2.2) by $\tilde{x}^{i+1}$ which is a locally maximum extreme solution with improved objective value $\varphi(\tilde{x}^{i+1},y^i)$. Note that $\tilde{x}^{i+1}$ can be determined by applying the bilinear programming algorithm (see § 4.1 of Part II).

(ii) The formulae of the cutting planes (Step 5)

The cutting plane of the form $g^t x \geq \tau_x$ can be generated by solving at most n parametric linear programming problems as before. On the other hand, the cutting plane of the form $g^t y \geq \tau_y$ necessitates the solution of parametric bilinear programming problems and will be ignored from the computational point of view. The determination of the cutting plane of the form $g^t x + h^t y \geq \tau$ is somewhat more difficult than in the case of BLP and may or may not be adjoined. It must be noted that the addition of the cut $g^t x \geq \tau_x$ at each locally maximum vertex by using $\varphi_o + \epsilon$ instead of $\varphi_o$ is sufficient to guarantee the finite convergence of the algorithm to an $\epsilon$-optimal solution if we choose the vector g appropriately.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Let us consider a QP which is more general than (5.2.1):

$$(5.2.3)\qquad \begin{cases} \text{maximize} & \psi(x,y) = c^t x + d^t y + \frac{1}{2} x^t P x + \frac{1}{2} y^t S y + x^t C y \\ \text{subject to} & Ex \leq e \ , \quad x \geq 0 \\ & Fy \leq f \ , \quad y \geq 0 \end{cases}$$

where $P \in R^{mxm}$ and $S \in R^{nxn}$ are both symmetric positive semi-definite matrices. The key Theorem 5.2.1 also holds in this case, but the extension of the BLP algorithm seems less promising since we have to solve a class of parametric bilinear programming problems to construct the cut of the form $g^t x \geq \tau_x$ or $h^t y \geq \tau_y$ .

5.2.2 The Bilinear Programming Problem with Coupling Constraints

Let us consider a quadratic programming problem

$$(5.2.4)\qquad \begin{cases} \text{maximize} & \varphi(x,y) = c^t x + d^t y + x^t C y \\ \text{subject to} & A_1 x + A_2 y \leq b \\ & x \geq 0 \ , \ y \geq 0 \end{cases}$$

This is another very important class of non-concave quadratic programming problem since it has a wide area of applications related to integer programming problems, game theoretic problems, etc But this is a far more difficult problem than the BLP (1.3) since the fundamental Theorem 2.1.1 does not hold for (5.2.4) and the BLP cannot be applied directly. Here we will suggest an algorithm which hopefully will generate a good solution.

Let $(x^o, y^o)$ be a feasible solution of (5.2.4). Given the $i^{th}$ feasible solution $(x^i, y^i)$ we will calculate the $i+1$-st feasible



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

solution $(x^{i+1},y^{i+1})$ by solving a pair of linear programs, i.e.,

$$
\begin{aligned}
x^{i+1} &= \arg\max \{\varphi(x,y^i) \mid A_1x \le b - A_2y^i,\ x \ge 0\} \\
y^{i+1} &= \arg\max \{\varphi(x^{i+1},y) \mid A_2y \le b - A_1x^{i+1},\ y \ge 0\}
\end{aligned} \tag{5.2.5}
$$

Obviously $\varphi(x^{i+1},y^i) \ge \varphi(x^i,y^i)$ since $x^i$ is a feasible solution of the linear program defining $x^{i+1}$. Similarly, we have $\varphi(x^{i+1},y^{i+1}) \ge \varphi(x^{i+1},y^i)$. We will proceed in this manner until we have $|\varphi(x^{k+1},y^{k+1}) - \varphi(x^k,y^k)| < \epsilon$ for some predetermined positive constant $\epsilon$. When this condition is obtained, let $x^\infty = x^{k+1}$ and $y^\infty = y^{k+1}$ and solve the BLP:

$$
\begin{cases}
\text{maximize} & \varphi(x,y) = c^tx + d^ty + x^tCy \\
\text{subject to} & A_1x \le b_1\ ,\ x \ge 0 \\
& A_2y \le b_2\ ,\ y \ge 0
\end{cases} \tag{5.2.6}
$$

where $b_1 = b - A_2y^\infty$ and $b_2 = b - A_1x^\infty$. Let $(x^*(b_1),y^*(b_2))$ be an optimal (or an $\epsilon$-optimal solution) of (5.2.6). Obviously, $\varphi(x^*(b_1),y^*(b_2)) \ge \varphi(x^\infty,y^\infty)$ since $(x^\infty,y^\infty)$ is a feasible solution of (5.2.6). Moreover, $(x^*(b_1),y^*(b_2))$ is a feasible solution of (5.2.4). If $(x^*(b_1),y^*(b_2)) = (x^\infty,y^\infty)$, then we cannot proceed any more by this process and will stop. If, on the other hand, $(x^*(b_1),y^*(b_2)) \ne (x^\infty,y^\infty)$ we will restart the whole process again by taking $(x^*(b_1),y^*(b_2))$ as a starting point. If $|\varphi(x^{k+1},y^{k+1}) - \varphi(x^k,y^k)| < \epsilon$ and $|\varphi(x^{k+1},y^{k+1}) - \varphi(x^*(b_1),y^*(b_2))| < \epsilon$, then we will stop.

Obviously this process does not guarantee the convergence to an optimal solution, but it will generate a 'good' feasible solution.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

PART II

Applications of Bilinear Programs



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

## INTRODUCTION

In this part of the paper, we will discuss some applications of the bilinear programming problem:

$$\begin{cases} \text{maximize} & \varphi(x,y) = c^t x + d^t y + x^t C y \\ \text{subject to} & x \in X \,,\ y \in Y \end{cases}$$

where $x \in R^m$, $y \in R^n$, $c \in R^m$, $d \in R^n$, $C \in R^{m \times n}$ and $X \subset R^m$, $Y \subset R^n$ are the polyhedral convex sets.

The structure of this problem might look rather special, but there actually exist many problems which are directly or indirectly formulated in this form. Here we will pick up several typical examples and look into the application of the algorithm developed in Part I. Problems to be discussed are:

1. Game-theoretic Problems (Chapter 1)
   (a) Equilibrium point of a constrained bimatrix game ... a bimatrix game with side constraints on the choice of mixed strategies of each player.
   (b) Two-move game with perfect information ... a zero-sum two person-two stage game in which the second player chooses his mixed strategy to maximize his income given the complete information of the first player's mixed strategy chosen in the first stage.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

2. Finite-horizon Markovian Decision Problems (Chapter 2)

   (a) Multi-stage Markovian assignment problem ... a multi-stage assignment problem in which the outcome associated with a certain assignment depends upon the assignment in the previous stage.

   (b) Multi-stage sales optimization of a monopolistic enterprise ... an optimal sales schedule of a monopolistic enterprise under the market situation that the unit price of the commodity depends upon the amount of supply in the previous stage.

3. Complementary (Orthogonal) Planning Problems (Chapter 3)

   (a) Complementary flows in a network ... a two commodity flow problem in which two different types of commodities cannot share the same arc.

   (b) Orthogonal production scheduling ... an optimal multi-stage production schedule under the condition that some of the activities cannot be activated in two consecutive stages.

4. Reduction of a Certain Class of Quadratic Programming Problems to Bilinear Programming Problems (Chapter 4)

   (a) Maximization of a convex quadratic function over a polyhedral convex set ... Reformulation of the problem as the equivalent bilinear programming problem and a comparison of the depth of cuts with those obtained by other methods.

   (b) 0-1 integer programming problems ... Application of the BLP algorithm to the 0-1 integer program.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

5. Miscellaneous Bilinear Programming Problems (Chapter 5)

(a) Reduction of a sparse matrix into an almost-triangular matrix by row and column permutations.

(b) A location problem on a rectangular network ... The simultaneous determination of (i) the location of a factory on a rectangular network and (ii) the amount of n types of raw materials to be shipped to the factory from m neighboring cities each having different capacity of supply and different unit price for n types of raw materials, in such a way as to minimize the total purchasing and transportation cost.

This class of models seem to have very wide applications in the real world situations but none appear to have been treated in the literature or solved except by the complete enumeration technique. More detailed analysis for these problems than the ones developed in this paper would be worthwhile in the future.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

## 1. GAME THEORETIC APPLICATIONS

The structure of the bilinear programming problem suggests a class of applications with game-theoretic flavor. Among these we will select two typical examples. One is an extension of the standard bimatrix game [N-1] and the other is an alternate formulation of a two-move game treated by G. Dantzig [D-2].

### 1.1 An Equilibrium Point of a Constrained Bimatrix Game

The standard bimatrix game [N-1] with players $P_1$ and $P_2$ is defined as follows. Let $S = \{s_1,\ldots,s_m\}$ and $T = \{t_1,\ldots,t_n\}$ be the set of alternatives available to $P_1$ and $P_2$, respectively. Assume that the payoffs to $P_1$ and $P_2$ when $P_1$ chooses $s_i$ and $P_2$ chooses $t_j$ are given by $a_{ij}$ and $b_{ij}$, respectively. Let $x = (x_1,\ldots,x_m)^t$ and $y = (y_1,\ldots,y_n)^t$ be the mixed strategies of $P_1$ and $P_2$, i.e., $x \in X_o$ and $y \in Y_o$ where

$$
\begin{aligned}
X_o &= \{x \in R^m \mid e_m^t x = 1 \ , \ x \geq 0\} \\
Y_o &= \{y \in R^n \mid e_n^t y = 1 \ , \ y \geq 0\}
\end{aligned}
\tag{1.1.1}
$$

where $e_m$ and $e_n$ are vectors of ones in $R^m$ and $R^n$. The expected payoffs to $P_1$ and $P_2$ when $P_1$ chooses $x$ and $P_2$ chooses $y$ are given by $x^tAy$ and $x^tBy$, respectively, where $A = (a_{ij})$, $B = (b_{ij})$.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Definition 1.1.1. A pair of mixed strategies $(x^*, y^*) \in X_o \times Y_o$ is called a Nash equilibrium point of a bimatrix game if

$$(1.1.2) \qquad \begin{cases} (x^*)^t A y^* = \max \{x^t A y^* \mid x \in X_o\} \\ (x^*)^t B y^* = \max \{(x^*)^t B y \mid y \in Y_o\} \end{cases}$$

It is well known that the problem of obtaining a Nash equilibrium point of a bimatrix game is reducible to a special case of the linear complementarity problem (LCP) defined below:

$$(LCP) \qquad \begin{cases} \text{Given } s \in R^{\ell} \text{ and } M \in R^{\ell \times \ell}, \text{ find } z \in R^{\ell} \text{ such that} \\ \\ z \geq 0 \ , \ s + Mz \geq 0 \ , \ z^t(s + Mz) = 0 . \end{cases}$$

where $z$, $s$, $M$ associated with (1.1.2) are given by

$$z = \begin{pmatrix} x \\ \\ y \end{pmatrix} \ , \quad s = \begin{pmatrix} -e_m \\ \\ -e_n \end{pmatrix} \ , \quad M = \begin{pmatrix} 0 & A \\ \\ B^t & 0 \end{pmatrix}$$

The 'complementary pivot algorithm' constructed by C. E. Lemke and J. T. Howson, Jr.([L-2],[C-5]) generates an equilibrium point of a bimatrix game in finitely many steps.

Now let us consider a 'constrained' bimatrix game in which players $P_1$ and $P_2$ are subject to more general constraints on the choice of their mixed strategies. Let $X$ and $Y$ be the sets of admissible strategies for $P_1$ and $P_2$, respectively. We will denote this constrained game by $\Gamma(A,X;B,Y)$ .



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Definition 1.1.2. A pair of 'admissible' strategies $(x^*,y^*) \in X \times Y$ is called a Nash equilibrium point of $\Gamma(A,X;B,Y)$ if

$$(x^*)^t Ay^* = \max \{x^t Ay^* \mid x \in X\}$$

$$(x^*)^t By^* = \max \{(x^*)^t By \mid y \in Y\}$$

The following fundamental theorem has been proved by Nash [N-1].

Theorem 1.1.1. If both $X$ and $Y$ are non-empty, closed, bounded and convex, then $\Gamma(A,X;B,Y)$ has a Nash equilibrium point for all $A$ and $B$.

Let us specialize our sets of admissible strategies

$$(1.1.3) \qquad X = X_1 \equiv \{x \mid Px \leq p\ ,\ e_m^t x = 1\ ,\ x \geq 0\} \neq \emptyset$$

$$Y = Y_1 \equiv \{y \mid Qy \leq q\ ,\ e_n^t y = 1\ ,\ y \geq 0\} \neq \emptyset$$

Obviously this choice of $X$ and $Y$ satisfies the conditions of Theorem 1.1.1, so there always exists an equilibrium point.†

In 1960, H. Mills [M-3] showed a way to reduce an equilibrium point problem of $\Gamma(A,X_o;B,Y_o)$ to a bilinear programming problem and in 1964, O. Mangasarian and H. Stone [M-2] proposed the application of Rosen's gradient projection method [R-4] and Balinski's method (for enumerating all the extreme points of a polyhedral convex set) [B-2]. Though their computational propositions are less attractive than the beautiful complementary pivot algorithm, their reduction technique applies as well to

†If $A = -B$, i.e., if the game is zero sum, then A. Charnes [C-2] showed that the problem is reducible to a pair of linear programs.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$\Gamma(A,X_1;B,Y_1)$ . We will follow this line and propose the application of our bilinear programming algorithm.

Theorem 1.1.2. A necessary and sufficient condition for $(x^*,y^*)$ to be an equilibrium point of $\Gamma(A,X_1;B,Y_1)$ is that it is the subvector of an optimal solution of the following bilinear programming problem

$$
(1.1.4)\quad
\begin{cases}
\text{maximize } \varphi(x,x_{m+1},y,y_{n+1},u,v) \\
\qquad = x^t(A+B)y - x_{m+1} - p^tu - y_{n+1} - q^tv \\
\text{subject to } B^tx - q^tv - x_{m+1}e_n \le 0 \\
\qquad Px \le p \\
\qquad e_m^tx = 1 \\
\qquad Ay - P^tu - y_{n+1}e_m \le 0 \\
\qquad Qy \le q \\
\qquad e_n^ty = 1 \\
\qquad x \ge 0 \ , \ u \ge 0 \ , \ y \ge 0 \ , \ v \ge 0
\end{cases}
$$

The values of $p^tu + y_{n+1}$ and $q^tv + x_{m+1}$ at the maximum $p^tu^* + y_{n+1}^*$ and $q^tu^* + x_{m+1}^*$ equal the expected payoff to the players $P_1$ and $P_2$, respectively. Also, $\varphi(x^*,x_{m+1}^*,y^*,y_{n+1}^*,u^*,v^*) = 0$ .

Proof. Let $z = (x,x_{m+1},y,y_{n+1},u,v)$ and $Z = \{z \mid z$ satisfies the constraints of (1.1.4)$\}$.

Let $(x^*,y^*)$ be an equilibrium point of $\Gamma(A,X_1;B,Y_1)$ . Then by definition,

$$
(1.1.5)\quad
\begin{aligned}
(x^*)^tAy^* &= \max \{x^tAy^* \mid Px \le p \ , \ e_m^tx = 1 \ , \ x \ge 0\} \\
(x^*)^tBy^* &= \max \{(x^*)^tBy \mid Qy \le q \ , \ e_n^ty = 1 \ , \ y \ge 0\}
\end{aligned}
$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The right hand side of (1.1.5) are linear programming problems whose constraint sets are bounded. By the duality theorem, $(x^*,y^*)$ is an equilibrium point of $\Gamma(A,X_1;B,Y_1)$ if and only if there exist a set of multipliers $u^*,v^*,x^*_{m+1}$ and $y^*_{n+1}$ such that

$$
(1.1.6)\qquad
\begin{cases}
(x^*)^t A y^* - p^t u^* - y_{n+1} = 0 \\
A y^* - P^t u^* - y_{n+1} e_m \le 0 \\
P x^* \le p \\
e_m^t x^* = 1 \\
x^* \ge 0 \quad , \quad u^* \ge 0 \\
(x^*)^t B y^* - q^t v^* - x_{m+1} = 0 \\
B^t x^* - Q^t v^* - x_{m+1} e_n \le 0 \\
Q y^* \le q \\
e_n^t y^* = 1 \\
y^* \ge 0 \quad , \quad v^* \ge 0
\end{cases}
$$

(i) Necessity. Let $z^* = (x^*,x^*_{m+1},y^*,y^*_{n+1},u^*,v^*)$ be a solution of the system (1.1.6). Obviously $z^* \in Z$ and $\varphi(z^*) = 0$. Premultiplying $y^t \ge 0$ and $x^t \ge 0$ to the first and the fourth inequalities of (1.1.4), we have

$$
(1.1.7)\qquad
\begin{aligned}
y^t B^t x - y^t Q^t v - x_{m+1} y^t e_n \le 0 \\
x^t A y - x^t P^t u - y_{n+1} x^t e_m \le 0
\end{aligned}
$$

Using $Px \le p$, $Qy \le q$, $e_m^t x = 1$ and $e_n^t y = 1$, these inequalities reduce to

$$
(1.1.8)\qquad
\begin{aligned}
x^t B y - q^t v - x_{m+1} \le 0 \\
x^t A y - p^t u - y_{n+1} \le 0
\end{aligned}
$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

These inequalities imply $\varphi(z) \leq 0 = \varphi(z^*)$ for all $z \in Z$. Hence $z^*$ is an optimal solution of (1.1.4).

(ii) Sufficiency. Let $z^* = (x^*, x^*_{m+1}, y^*, y^*_{n+1}, u^*, v^*)$ be an optimal solution of the bilinear programming problem (1.1.4). Then by (1.1.8), $\varphi(z^*) \leq 0$. By Theorem 1.1.1, there exists an equilibrium point $(\hat{x},\hat{y})$ together with multipliers $\hat{u}$, $\hat{v}$, $\hat{x}_{m+1}$, $\hat{y}_{n+1}$ satisfying (1.1.6). It follows that $\hat{z} = (\hat{x},\hat{x}_{m+1},\hat{y},\hat{y}_{n+1},\hat{u},\hat{v}) \in Z$ and $\varphi(\hat{z}) = 0$. So $\varphi(z^*) = 0$. This holds if and only if $(x^*)^t A y^* - p^t u^* - y_{n+1} = 0$ and $(x^*)^t B y^* - q^t v^* - x^*_{m+1} = 0$. Hence $z^*$ satisfies (1.1.6), so that $z^*$ is an equilibrium point of $\Gamma(A,X_1;B,Y_1)$. The remaining statements follow immediately from (1.1.6). Q.E.D.

Theorem 1.1.2 shows that the solution of the bilinear programming problem (1.1.4) gives an equilibrium point of $\Gamma(A,X_1;B,Y_1)$. Moreover, we know a priori that the optimal objective value $\varphi^*$ is zero. When we apply our bilinear programming algorithm this property will enhance its efficiency remarkably: In addition to the fact that we can quit computation if we attain zero, the cuts to be introduced at a non optimal locally maximum vertex $z^p$ will turn out to be very deep ones especially when that vertex is a 'bad' vertex in the sense that $\varphi(z^p)$ is far below zero.

Now let us turn to the alternative formulation of the same problem as a linear complementarity problem (LCP)



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Theorem 1.1.3. A necessary and sufficient condition for $(\bar{x},\bar{y})$ to be an equilibrium point of $\Gamma(A,X_1;B,Y_1)$ is that $\bar{z}^t = (\bar{x}^t,\bar{v}^t,\bar{\lambda}_1,\bar{\lambda}_2,\bar{y}^t,\bar{u}^t,\bar{\mu}_1,\bar{\mu}_2)$ be a solution of a LCP where

$$M = \left[\begin{array}{c|cc|cc} & -A & P^t & e_m & -e_m \\ & -Q & & & \\ 0 & & 0 & & \\ & e_n^t & & & \\ & -e_n^t & & & \\ \hline -B^t & Q^t & e_n & -e_n & \\ -P & & & & \\ & 0 & & 0 & \\ e_m^t & & & & \\ -e_m^t & & & & \end{array}\right], \quad s = \begin{bmatrix} 0 \\ q \\ -1 \\ 1 \\ 0 \\ p \\ -1 \\ 1 \end{bmatrix}$$

Proof. Follows directly from (1.1.6).

Unfortunately, however, the validity of the complementary pivot algorithm depends heavily upon the property of the matrix $M$ and it is not clear whether this algorithm will generate a complementary solution for $M$ defined above (see [E-1])†.

†Very recently, Professor B. C. Eaves communicated to the author that he proved that the complementary pivot algorithm generates an equilibrium point of a 'constrained polymatrix game' of which a 'constrained bimatrix game' is a special case.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

## 1.2 Two-Move Game with Perfect Information

In this section, we will consider a two-move game first introduced by G. Dantzig [D-2] and show that this problem can be reformulated as a bilinear programming problem.

Let there be two enterprises $P_1$ and $P_2$. Let $S = \{s_1, \ldots, s_m\}$ and $T = \{t_1, \ldots t_n\}$ be the set of activities available to $P_1$ and $P_2$, respectively. At the first stage of the game, $P_1$ chooses an activity level $x = (x_1, \ldots, x_m)^t \geq 0$ as long as it is consistent with a given inventory level $b = (b_1, \ldots b_r)^t$. Assume that $P_1$'s constraint set $X$ is given by

$$X = \{x \mid A_1 x \leq b \ , \ x \geq 0\} \tag{1.2.1}$$

where $A_1 \in R^{r \times m}$. $X$ is called a set of $P_1$'s admissible moves. As a result of the move $x \in X$, $P_1$ leaves an inventory of level $d + Cx$ for the enterprise $P_2$ where both $d = R^{\ell}$ and $C \in R^{\ell \times m}$ are known quantities. At the second stage of the game, $P_2$ will choose an activity vector $y = (y_1, \ldots y_n)^t \geq 0$ from the set $Y(x)$ of admissible moves. Let $Y(x)$ be given by

$$Y(x) = \{y \mid A_2 y \leq d + Cx \ , \ y \geq 0\} \tag{1.2.2}$$

where $A_2 \in R^{s \times n}$. For simplicity, $Y(x)$ is assumed to be non-empty and bounded for all $x \in X$.

Let us assume that the payment $f(x,y)$, from $P_1$ to $P_2$, when $P_1$ and $P_2$ choose $x$ and $y$, respectively, is given by

$$f(x,y) = p^t x + q^t y \tag{1.2.3}$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

where $p \in R^m$ and $q \in R^n$. Given $x$, $P_2$ naturally tries to maximize $f(x,y)$ over $y \in Y(x)$, so his problem is to solve

$$\text{(1.2.4)} \qquad \text{maximize } \{q^t y \mid y \in Y(x)\}$$

By assumption, this problem has an optimal solution $y(x)$ for all $x \in X$. On the other hand, $P_1$ will choose $x$ from $X$ so as to minimize $f(x,y(x))$ and the problem for $P_1$ is to solve

$$\text{(1.2.5)} \quad \begin{cases} \text{minimize} & f(x,y(x)) = p^t x + q^t y(x) \\ \text{subject to} & x \in X \end{cases}$$

By the duality theorem of linear programming, we have

$$\begin{aligned} q^t y(x) &= \max_y \{q^t y \mid A_2 y \le d + Cx,\ y \ge 0\} \\ &= \min_z \{(d + Cx)^t z \mid A_2^t z \ge q,\ z \ge 0\} \end{aligned}$$

so (1.2.5) becomes

$$\text{(1.2.6)} \quad \begin{cases} \text{minimize}_x & [p^t x + \min_z \{(d + Cx)^t z \mid A_2^t z \ge q,\ z \ge 0\}] \\ \text{subject to} & A_1 x \le b,\ x \ge 0 \end{cases}$$

or equivalently

$$\text{(1.2.7)} \quad \begin{cases} \text{minimize}_{x,z} & \varphi(x,z) = d^t z + p^t x + z^t C x \\ \text{subject to} & -A_2^t z \le -q,\ z \ge 0 \\ & A_1 x \le b,\ x \ge 0 \end{cases}$$

which is exactly a bilinear programming problem.

If we have an $\epsilon$-optimal solution of the BLP (1.2.7), then we can recover an $\epsilon$-optimal solution of (1.2.5) (which is the dual of (1.2.7)) by applying Theorem 1.1 of Part I.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Remark. Although we have stated the problem relative to the game played between two enterprises, the problems of the form (1.2.5) naturally arise also in the area of quality control, blending problem under uncertainty and many others, which will be discussed elsewhere.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

## 2. FINITE HORIZON MARKOVIAN DECISION PROBLEMS

It sometimes happens in formulating a multi-stage dynamic version of a single stage linear programming model that the decision at stage $t$ affects the unit cost (or unit profit) of certain activities at stage $t + 1$ (i.e., Markovian dependence). In this case the problem can no longer be formulated as a linear programming problem and we get into serious trouble. In this chapter, we will show that some of these types of problems can be formulated as a bilinear programming problem. The examples to be discussed will be a multi-stage Markovian assignment problem and a problem related to a monopoly system, neither of which appears to have been treated in the literature.

### 2.1 Multi-Stage Markovian Assignment Problem

A standard assignment problem is defined as follows: Let there be $N$ machines and $N$ jobs and assume that one and only one machine has to be assigned to each of the $N$ jobs. Let $p_{ij}$ be the profit associated with assigning machine $i$ to job $j$. Introducing the binary variables

$$x_{ij} = \begin{cases} 1 & \text{if we assign machine } i \text{ to job } j \\ 0 & \text{otherwise} \end{cases}$$

this problem is formulated as



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$
(2.1.1)\qquad
\begin{cases}
\text{maximize} & \displaystyle\sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}x_{ij} \\
\text{subject to} & \\
& \displaystyle\sum_{j=1}^{N} x_{ij} = 1 \qquad i=1,\ldots,N. \\
& \displaystyle\sum_{i=1}^{N} x_{ij} = 1 \qquad j=1,\ldots,N. \\
& x_{ij} = 0 \text{ or } 1\,, \quad i,j=1,\ldots,N.
\end{cases}
$$

It is well known that this problem is equivalent to the following linear program:

$$
(2.1.2)\qquad
\begin{cases}
\text{maximize} & \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \\
\text{subject to} & \\
& \displaystyle\sum_{j=1}^{N} x_{ij} = 1 \qquad i=1,\ldots,N \\
& \displaystyle\sum_{i=1}^{N} x_{ij} = 1 \qquad j=1,\ldots,N \\
& x_{ij} \geq 0 \qquad i,j=1,\ldots,N
\end{cases}
$$

which can be solved very efficiently by a variant of the simplex method [D-1] or by network-flow theory [F-1].



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Now let us consider the following $L$ stage dynamic version of this problem. We will assume for simplicity that the number of machines and the number of jobs are fixed at $N$ for any stage $\ell$, $\ell=1,\ldots,L$. As before, one and only one machine has to be assigned to each of $N$ jobs at every stage. Further, let us assume that the value associated with an assignment of machine $i$ to job $j$ at stage $\ell$ depends on whether or not this machine was assigned to job $k$ at stage $\ell-1$ and if it was assigned the outcome is given by $p^{\ell}_{ij} + q^{\ell}_{ijk}(q^{1}_{ijk} \equiv 0$ for all $i,j)$.

As before, we will introduce the binary variables

$$x^{\ell}_{ij} = \begin{cases} 1 & \text{if machine } i \text{ is assigned to job } j \text{ at stage } \ell \\ 0 & \text{otherwise} \end{cases}$$

and let $x^{\ell} = (x^{\ell}_{11}, x^{\ell}_{12}, \ldots, x^{\ell}_{1N}, x^{\ell}_{21}, \ldots, x^{\ell}_{NN})^{t}$. Then the total outcome $s^{\ell}(x^{\ell}|x^{\ell-1})$ at period $\ell$ when we choose $x^{\ell}$ given $x^{\ell-1}$ is:

$$s^{\ell}(x^{\ell}|x^{\ell-1}) = \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}(p^{\ell}_{ij}+q^{\ell}_{ijk})x^{\ell}_{ij}x^{\ell-1}_{ik}$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{N} p^{\ell}_{ij}x^{\ell}_{ij} + \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} q^{\ell}_{ijk}x^{\ell}_{ij}x^{\ell-1}_{ik}$$

So the total outcome $S^{L}(x^{1},\ldots,x^{L})$ through $L$ stages associated with the choice of vectors $x^{1},\ldots,x^{L}$ is given by

$$S^{L}(x^{1},\ldots,x^{L}) = \sum_{\ell=1}^{L} s^{\ell}(x^{\ell}|x^{\ell-1})$$

$$= \sum_{\ell=1}^{N}\sum_{i=1}^{N}\sum_{j=1}^{N} p^{\ell}_{ij}x^{\ell}_{ij} + \sum_{\ell=2}^{N}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} q^{\ell}_{ijk}x^{\ell}_{ij}x^{\ell-1}_{ik}$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

and our problem becomes

$$
(2.1.3)\quad
\begin{cases}
\text{maximize } S^{L}(x^{1},\ldots,x^{L}) = \displaystyle\sum_{\ell=1}^{L}\sum_{i=1}^{N}\sum_{j=1}^{N} p_{ij}^{\ell} x_{ij}^{\ell} \\
\qquad\qquad + \displaystyle\sum_{\ell=2}^{L}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} q_{ijk}^{\ell} x_{ij}^{\ell} x_{ik}^{\ell-1} \\
\text{subject to} \quad \displaystyle\sum_{j=1}^{N} x_{ij}^{\ell} = 1 \quad , \quad i=1,\ldots,N \quad ; \\
\displaystyle\sum_{i=1}^{N} x_{ij}^{\ell} = 1 \quad , \quad j=1,\ldots,N \quad ; \quad \ell=1,\ldots,L \quad , \\
x_{ij}^{\ell} = 0 \text{ or } 1 \quad , \quad i,j=1,\ldots,N \quad .
\end{cases}
$$

Let

$$
(2.1.4)\quad Q^{\ell} = \begin{bmatrix} Q_{1}^{\ell} & & & 0 \\ & Q_{2}^{\ell} & & \\ & & \ddots & \\ 0 & & & Q_{N}^{\ell} \end{bmatrix} , \quad p^{\ell} = \begin{bmatrix} p_{11}^{\ell} \\ \vdots \\ p_{21} \\ \vdots \\ p_{NN}^{\ell} \end{bmatrix} \quad ; \quad \ell=1,\ldots,L
$$

where

$$
(2.1.5)\quad Q_{i}^{\ell} = \begin{bmatrix} q_{i11}^{\ell} & q_{i12}^{\ell} & \cdots & q_{i1N}^{\ell} \\ q_{i21}^{\ell} & q_{i22}^{\ell} & & q_{i2N}^{\ell} \\ \vdots & & \ddots & \vdots \\ q_{iN1}^{\ell} & q_{iN2}^{\ell} & \cdots & q_{iNN}^{\ell} \end{bmatrix} \quad ; \quad i=1,\ldots,N \quad ; \quad \ell=1,\ldots,L \quad .
$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

and let

$$X = \left\{ x=(x_{11},x_{12}\cdots,x_{1N},x_{21},\cdots,x_{NN})^t \;\middle|\; \sum_{j=1}^{N} x_{ij} = 1, \quad i=1,\cdots,N \; ; \right.$$

(2.1.6)

$$\left. \sum_{i=1}^{N} x_{ij} = 1 \; , \quad j=1,\cdots,N \; ; \quad x_{ij} = 0 \;\text{ or }\; 1, \; i,j=1,\cdots,N \right\}$$

then the problem is rewritten as

$$\begin{cases} \text{maximize} \quad \sum_{\ell=1}^{L} (p^{\ell})^t x^{\ell} + \sum_{\ell=2}^{L} (x^{\ell})^t Q^{\ell} x^{\ell-1} \\ \text{subject to} \quad x^{\ell} \in X \; , \quad \ell=1,\cdots,L \end{cases} \tag{2.1.7}$$

Let us assume for simplicity that $L$ is even and let

$$u = \begin{pmatrix} x^1 \\ x^3 \\ \cdot \\ \cdot \\ \cdot \\ x^{L-1} \end{pmatrix} , \; v = \begin{pmatrix} x^2 \\ x^4 \\ \cdot \\ \cdot \\ \cdot \\ x^{L} \end{pmatrix}, \quad c = \begin{pmatrix} p^1 \\ p^3 \\ \cdot \\ \cdot \\ \cdot \\ p^{L-1} \end{pmatrix}, \; d = \begin{pmatrix} p^2 \\ p^4 \\ \cdot \\ \cdot \\ \cdot \\ p^{L} \end{pmatrix} \tag{2.1.8}$$

$$Q = \begin{bmatrix} Q_2 & & & & 0 \\ Q_3^t & Q_4 & & & \\ & Q_5^t & Q_6 & & \\ & & \ddots & \ddots & \\ 0 & & & Q_{L-1}^t & Q_L \end{bmatrix}$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Then (2.1.7) further reduces to

$$(2.1.9)\qquad \begin{cases} \text{maximize} \quad \varphi(u,v) = c^t u + d^t v + u^t Q v \\ \\ \text{subject to} \quad u \in \prod_{i=1}^{L/2} X \ , \quad v \in \prod_{i=1}^{L/2} X \end{cases}$$

Hence by Theorem 2.1.1 of Part I, there exist an optimal solution $(u^*, v^*)$ such that $u^* \in \operatorname{ext} \prod_{i=1}^{L/2} X$ and $v^* \in \operatorname{ext} \prod_{i=1}^{L/2} X$, i.e., there exists an optimal solution $x^* = (x^{1*}, \ldots, x^{L*})$ such that $x^{\ell *} \in \operatorname{ext} X$.

By the fundamental theorem of G. Birkhoff [D-1], ext X = ext $X_o$ where

$$X_o = \left\{ x = (x_{11}, \ldots x_{1N}, x_{21}, \cdots, x_{NN}) \;\middle|\; \sum_{j=1}^{N} x_{ij} = 1 \ , \quad i=1,\ldots,N \ ; \right.$$

$$\left. \sum_{i=1}^{N} x_{ij} = 1 \ , \ j=1,\ldots,N; \quad x_{ij} \geq 0 \ , \ i,j=1,\ldots,N \right\}$$

Hence (2.1.8) is equivalent to

$$(2.1.10)\qquad \begin{cases} \text{maximize} \quad \varphi(u,v) = c^t u + d^t v + u^t Q v \\ \\ \text{subject to} \quad u \in \prod_{i=1}^{L/2} X_o \ , \quad v \in \prod_{i=1}^{L/2} X_o \end{cases}$$

which is, by definition, a bilinear programming problem.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

## 2.2 Multi-Stage Production and Sales Optimization of a Monopolistic Enterprise

Consider a decision problem of an enterprise in the following market situation. This enterprise is producing $n$ types of commodities $j=1,\ldots,n$ and it is a dominating supplier of some of these commodities, say $j=1,\ldots,n_1$ where $n_1 \leq n$. Assume that $a_{ij}$ units of raw materials of type $i$, $i=1,\ldots,m$ is required to produce a unit amount of commodity $j$. Let $b_i^{\ell}$ be the amount of raw material $i$ available in period $\ell, \ell=1,\ldots,L$. Let $x_j^{\ell}$ be the amount of commodity $j$ to be produced in period $\ell$. Then $x^{\ell} = (x_1^{\ell},\ldots,x_n^{\ell})^t$ has to satisfy the constraint

$$Ax^{\ell} \leq b^{\ell}, \quad x^{\ell} \geq 0 \tag{2.2.1}$$

for $\ell=1,\ldots,L$, where $A = (a_{ij})$ and $b^{\ell} = (b_1^{\ell},\ldots,b_m^{\ell})^t$. We will assume further that the commodities produced in period $\ell$ have to be sold or discarded in that period because there is no storage (or a very high holding cost is incurred). If we can assume that the unit production cost and the market price of these commodities are fixed constants for all $\ell$, then it suffices to solve $L$ independent linear programming problems to obtain an optimal production schedule. But if we assume that the market prices $p_j^{\ell}$ of the commodities $j=1,\ldots,n_1$ at period $\ell$ are dependent upon the supply $y_j^{\ell}$ of this commodity at period $\ell-1$, then the problem can no longer be formulated as a linear programming problem. Let $y^{\ell} = (y_1^{\ell}\ldots,y_n^{\ell})^t$, $(y^o \equiv 0)$, $p^{\ell} = (p_1^{\ell},\ldots,p_n^{\ell})^t$. If



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

we assume specifically that

$$(2.2.2)\qquad p_j^{\ell}(y^{\ell-1}) = \begin{cases} p_j^{\ell} + \sum_{k=1}^{n_1} q_{jk}^{\ell} y_k^{\ell-1} & , \quad j=1,\ldots,n_1 \\ p_j^{\ell} & \quad j=n_1+1,\ldots,n \end{cases}$$

then the total profit $f^{\ell}(y^{\ell},x^{\ell}|y^{\ell-1})$ at period $\ell$ for given $y^{\ell},x^{\ell}$ and $y^{\ell-1}$ is

$$\begin{aligned} f^{\ell}(y^{\ell},x^{\ell}|y^{\ell-1}) &= \sum_{i=1}^{n} \left\{ p_j^{\ell}(y^{\ell-1})\, y_j^{\ell} - r_j^{\ell} x_j^{\ell} \right\} \\ &= \sum_{j=1}^{n} (p_j^{\ell} y_j^{\ell} - r_j^{\ell} x_j^{\ell}) + \sum_{j=1}^{n_1} \sum_{k=1}^{n_1} q_{jk}^{\ell}\, y_j^{\ell}\, y_k^{\ell-1} \\ &= (p^{\ell})^t y^{\ell} - (r^{\ell})^t x^{\ell} + (y^{\ell})^t\, Q^{\ell} y^{\ell-1} \end{aligned}$$

where $r_j^{\ell}$ is the unit production cost of commodity $j$ at period $\ell$,

$$Q^{\ell} = \begin{bmatrix} q_{11}^{\ell} & \cdots & q_{1n_1}^{\ell} & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ q_{n_1 1}^{\ell} & \cdots & q_{n_1 n_1}^{\ell} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad r^{\ell} = \begin{bmatrix} r_1^{\ell} \\ r_2^{\ell} \\ \vdots \\ r_n^{\ell} \end{bmatrix}, \quad p^{\ell} = \begin{bmatrix} p_1^{\ell} \\ p_2^{\ell} \\ \vdots \\ p_n^{\ell} \end{bmatrix}$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

To optimize $L$ periods production and sales schedule, we have to solve

$$(2.2.3)\quad \begin{cases} \text{maximize} & f^L(y^1,\ldots,y^L,x^1,\ldots,x^L) \\ & = \sum_{\ell=1}^{L} (p^\ell)^t y^\ell - \sum_{\ell=1}^{L} (r^\ell)^t x^\ell + \sum_{\ell=2}^{L} (y^\ell)^t Q^\ell y^{\ell-1} \\ \text{subject to} & Ax^\ell \le b^\ell \,,\ x^\ell \ge 0 \\ & 0 \le y^\ell \le x^\ell \qquad\qquad \ell=1,\ldots,L \end{cases}$$

If we assume further that $r^\ell \ge 0$ for all $\ell$, then we necessarily have $x^\ell = y^\ell$ for all $\ell$ and the problem simplifies to

$$(2.2.4)\quad \begin{cases} \text{maximize} & \varphi(x^1,\ldots,x^L) \\ & = \sum_{\ell=1}^{L} (p^\ell - r^\ell)^t x^\ell + \sum_{\ell=2}^{L} (x^\ell)^t Q^\ell x^{\ell-1} \\ \text{subject to} & Ax^\ell \le b^\ell \\ & x^\ell \ge 0 \qquad\qquad \ell=1,\ldots,L \end{cases}$$

This has almost the same structure as (2.1.7) of the previous section. Let us assume for simplicity that $L$ is even. If we define

$$u = \begin{bmatrix} x_1 \\ x_3 \\ \cdot \\ \cdot \\ \cdot \\ x_{L-1} \end{bmatrix}, \quad v = \begin{bmatrix} x_2 \\ x_4 \\ \cdot \\ \cdot \\ \cdot \\ x_L \end{bmatrix}, \quad c = \begin{bmatrix} p^1 - r^1 \\ p^3 - r^3 \\ \cdot \\ \cdot \\ p^{L-1} - r^{L-1} \end{bmatrix}, \quad d = \begin{bmatrix} p^2 - r^2 \\ p^4 - r^4 \\ \cdot \\ \cdot \\ p^L - r^L \end{bmatrix}$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$Q = \begin{bmatrix} Q^2 & & & 0 \\ (Q^3)^t & Q^4 & & \\ & \ddots & \ddots & \\ 0 & & (Q^{L-1})^t & Q^L \end{bmatrix}$$

Then (2.2.4) further reduces to

$$(2.2.5) \qquad \begin{cases} \text{maximize} & \Phi(u,v) = c^t u + d^t v + v^t Q u \\ \text{subject to} & u \in U \ , \quad v \in V \end{cases}$$

where

$$U = \{(x^1,\ldots,x^{L-1}) \mid Ax^1 \le b^1,\ x^1 \ge 0,\ldots,Ax^{L-1} \le b^{L-1},\ x^{L-1} \ge 0\}$$

$$V = \{(x^2,\ldots,x^L) \mid Ax^2 \le b^2,\ x^2 \ge 0,\ldots,Ax^L \le b^L,\ x^L \ge 0\} \qquad (2.2.6)$$

(2.2.5) is, by definition, a bilinear programming problem and is solvable by our algorithm.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

## 3. COMPLEMENTARY (ORTHOGONAL) PLANNING PROBLEMS

### 3.1 Introduction

As will be discussed in detail in the following sections, we are sometimes required to solve the system of the form

$$(3.1.1)\qquad \begin{cases} \text{maximize} \quad f(x,y) = c^t x + d^t y \\ \text{subject to} \quad Ex \leq e \qquad (x \in R^n) \\ \qquad\qquad\quad Fy \leq f \qquad (y \in R^n) \\ \qquad\qquad\quad x \geq 0\ ,\ y \geq 0 \end{cases}$$

with additional constraint

$$(3.1.2)\qquad x^t y = 0$$

where $x \in R^n$, $y \in R^n$, $c \in R^n$, $d \in R^n$, $E \in R^{k_1 \times n}$, $F \in R^{k_2 \times n}$. Let us call (3.1.2) a 'complementarity' constraint or an 'orthogonality' constraint. If the constraint set of (3.1.1) is bounded, then (3.1.2) together with non-negativity constraint is equivalent to

$$(3.1.3)\qquad \begin{cases} x \leq M_o u \\ y \leq M_o(e_n - u) \\ x \geq 0\ ,\ y \geq 0 \\ u_i = 0 \text{ or } 1\ , \quad i=1,\ldots,n \end{cases}$$

where $M_o$ is a large constant and $e_n$ is the n-dimensional vector of 1's (see [S-2]. So the system (3.1.1) - (3.1.2) is equivalent to the following mixed integer problem



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$
(3.1.4)\qquad
\begin{cases}
\text{maximize} & f(x,y) = c^t x + d^t y \\
\text{subject to} & Ex \leq e \\
& Fy \leq f \\
& x - M_o u \leq 0 \\
& y + M_o u \leq M_o e_n \\
& x \geq 0\ ,\ y \geq 0 \\
& u_i = 0 \text{ or } 1, \quad i=1,\ldots,n\ .
\end{cases}
$$

But this is a considerably larger system compared with the original one especially when $m \gg k_1$ or $n \gg k_2$ and the efficiency of a mixed integer algorithm [B-3] or branch and bound algorithm [B-1] applied to (3.1.4) is rather doubtful.

Instead, we will propose here the resolution of (3.1.2) by the application of a classical penalty function method. Let us consider the bilinear programming problem:

$$
(3.1.5)\qquad
\begin{cases}
\text{maximize} & \varphi_M(x,y) = c^t x + d^t y - Mx^t y \\
\text{subject to} & Ex \leq e \\
& Fy \leq f \\
& x \geq 0\ ,\ y \geq 0
\end{cases}
$$

where $M$ is a large constant. The equivalence of the system (3.1.1) - (3.1.2) to (3.1.5) will be established by the standard technique:

Theorem 3.1.1. If the constraint set of (3.1.1) is bounded, then there exists a constant $M_o$ such that (3.1.5) is equivalent to the system (3.1.1) - (3.1.2) for $M \geq M_o$ in the sense that an optimal solution of the former problem is also an optimal solution of the latter.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Though the high penalty in (3.1.5) may cause instability in actual computation, our algorithm seems to be more promising than solving the mixed integer problem (3.1.4) by a branch and bound algorithm since the latter is essentially the same thing as enumerating all possible combinations of the vectors $x$ and $y$ satisfying $x^t y = 0$ .

## 3.2 Complementary Flows in a Network

Let $G = (N,A)$ be a network where $N$ is the set of nodes and $A$ is the set of arcs. We will assume that $G$ is a simple network, i.e., it has no loops or multiple edges. Associated with each arc $a_i \in A$ are the constants $c_i$ and $p_i$ which represent the capacity of $a_i$ and the cost of a unit flow on $a_i$ . Assume further that there exist $k$ pairs of distinguished nodes $s_i$ and $t_i$ , $i=1,\ldots,k$ called sources and sinks, respectively. Let $S = \{s_1,\ldots,s_k\}$ and $T = \{t_1,\ldots,t_k\}$.

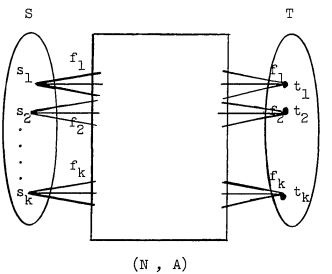



Figure 3.2.1



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The following 'single commodity' flow problems associated with this network have been considered and solved successfully [F-1].

(i) Feasibility Problem: Can we send a commodity of quantity $f$ from the set of sources $S$ to the set of sinks $T$ through this network $(N,A)$ ?

(ii) Max - Flow Problem: What is the maximum amount of flow $f$ we can send from $S$ to $T$ through $(N,A)$ ?

(iii) Minimum Cost Flow Problem: What is the minimum cost associated with the flow of value $f$ from $S$ to $T$ on $(N,A)$ ?

Based upon the successful resolution of these problems, the following extension has been considered: Instead of one commodity, let there be $k$ types of commodities and assume that we have to send $f_i$ units of commodity $i$ from $s_i$ to $t_i$, $i=1,\ldots,k$ . This is called a multi-commodity flow problem and the problems analogous to those described above have been considered but with less success [H-1], [T-2].

(i)' Feasibility Problem: Can we send $f_i$ units of commodity $i$ from $s_i$ to $t_i$ for $i=1,\ldots,k$ through $(N,A)$ ?

(ii)' Max - Flow Problem: What is the maximum of the sum of flows $f_i$ from $s_i$ to $t_i$ $i=1,\ldots,k$ through $(N,A)$ ?

(iii)' Minimum Cost Flow Problem: What is the minimum cost associated with flows $f_i$ from $s_i$ to $t_i$ $i=1,\ldots,k$ on $(N,A)$ ?

These are all considerably more difficult than their counterparts and the successful resolution comparable to the single commodity case has been made for the problems (i)' and (ii)' only when $k = 2$ [H-1].

Let us consider a 'complementary' version of a multi-commodity flow problem. We will assume that we have to send $f_i$ units of $i^{th}$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

commodity from $s_i$ to $t_i$ for i=1,...,k as before. It is further assumed that every two types of commodities are incompatible so that no two types of flows can share the same arc (consider the flows of chemicals, for example). For simplicity, let us first consider the easiest case, i.e., $k = 2$. In this case we will call the type of flow described above a 'complementary' flow in the sense that two types of commodities have to use the complementary set of arcs. Despite its obvious practical importance there does not appear to be a reference which has treated this kind of situation. Here we will consider the following three problems:

1. (Feasibility Problem). Does there exist a complementary flow of value $f_1$ and $f_2$ ?

2. (Maximum Complementary Flow Problem). What is the maximum of the sum $f_1 + f_2$ of the complementary flows?

Let $p^1_{ij}$ and $p^2_{ij}$ be the cost associated with unit flow of commodity I and II, respectively, on the arc connecting nodes i and j. Then the third problem will be stated as follows:

3. (Minimum Cost Complementary Flow Problem). What is the minimum cost associated with the complementary flows of values $f_1$ and $f_2$ ?

Let $x_{ij}$ and $y_{ij}$ be the amount of flows of commodity I and II, respectively, on the arc connecting the nodes i and j. Then, by the definition of a complementary flow, these quantities must satisfy

$$\begin{cases} x_{ij}y_{ij} = 0 \\ 0 \le x_{ij} \le c_{ij} \\ 0 \le y_{ij} \le c_{ij} \end{cases} \tag{3.2.1}$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

for all $i$ and $j \in S(i)$ where $S(i)$ is the set of nodes connected to node $i$ by some arc. By the flow conservation law, we have

$$(3.2.2) \qquad \sum_{i \in S(j)} x_{ij} - \sum_{k \in S(j)} x_{jk} = \begin{cases} -f_1 & \text{if } j = s_1 \\ 0 & \text{if } j \neq s_1, t_1 \\ f_1 & \text{if } j = t_1 \end{cases}$$

$$(3.2.3) \qquad \sum_{i \in S(j)} y_{ij} - \sum_{k \in S(j)} y_{jk} = \begin{cases} -f_2 & \text{if } j = s_2 \\ 0 & \text{if } j \neq s_2, t_2 \\ f_2 & \text{if } j = t_2 \end{cases}$$

Let us define the node-arc incidence matrix $P$ associated with this network in a standard manner and define the vectors $x, y, b^1, b^2$ and $c$ appropriately. Then (3.2.2) and (3.2.3) can be rewritten in a compact form

$$(3.2.4) \qquad \begin{aligned} Px &= b^1 \quad , \quad 0 \leq x \leq c \\ Py &= b^2 \quad , \quad 0 \leq y \leq c \end{aligned}$$

Let

$$(3.2.5) \qquad \begin{aligned} X &= \{x \mid Px = b^1 \, , \; 0 \leq x \leq c\} \\ Y &= \{y \mid Py = b^2 \, , \; 0 \leq y \leq c\} \end{aligned}$$

1. Feasibility Problem

This is equivalent to the problem of finding $x \in X$ and $y \in Y$ such that $x^t y = 0$ . Obviously this is reducible to the following bilinear programming problem:



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$(3.2.6) \qquad \begin{cases} \text{minimize} \quad \varphi_o(x,y) = x^t y \\ \text{subject to} \quad Px = b^1 \ , \quad 0 \le x \le c \\ \qquad\qquad\quad Py = b^2 \ , \quad 0 \le y \le c \end{cases}$$

If the minimand $\varphi_o(x,y)$ attains zero, then the corresponding $x$ and $y$ give a feasible complementary flow. If, on the other hand, the minimum of $\varphi_o(x,y)$ is positive, then there is no feasible complementary flow. This implies that we can use the value $0$ as a substitute of $\varphi_o$ through all the steps of the bilinear programming algorithm and the finite termination of the algorithm is guaranteed as in the problem associated with an equilibrium point of a (constrained) bimatrix game. Moreover, if the vectors $b^1, b^2$ and $c$ are vectors of intergers, then by the unimodularity of the basis matrix of $P$, every extreme solution has to have integer components. So if $\varphi(x,y) < 1$ is realized at some stage of the computation, then by the algorithm suggested in the proof of Theorem 2.1.1 of Part I, we can directly obtain an optimal solution $\varphi(x,y) = 0$ (i.e., a feasible complementary flow) by solving a pair of linear programs.

2. Maximum Complementary Flow Problem

Though this problem can be solved as a special case of the minimal cost complementary flow problem to be discussed shortly, it seems to be very difficult to construct an efficient algorithm for this problem as in the case of the standard (two-commodity) maximum flow problem and we will only state here the following rather obvious extension of a max-flow min-cut theorem for a single commodity network flow [F-1].



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Let $A_1$ and $A_2$ be the partition of the set of arcs $A$ into two mutually exclusive sets of arcs and let us define the 'complementary' networks $(N,A_1)$ and $(N,A_2)$. Let $c_i(A_i)$, i=1,2 be the minimal cut capacity separating $s_i$ and $t_i$ in $(N,A_i)$.

Theorem 3.2.1. (Max-Flow Max-Min Complementary Cut Theorem)

$$\max(f_1 + f_2) = \max_{A_1,A_2} \{c_1(A - A_2) + c_2(A - A_1)\} \tag{3.2.7}$$

Proof. Let $(x^*,y^*)$ be a maximal complementary flow in $(N,A)$, and let $f_1^*$ and $f_2^*$ be the flow values of commodity I and II corresponding to $x^*$ and $y^*$. Let $A_1^*$ and $A_2^*$ be, respectively, the sets of arcs on which a positive flow of commodity I and II exists. Then by definition $A_1^* \bigcap A_2^* = \emptyset$. Let $G_1^* = (N,A-A_2^*)$ and $G_2^* = (N,A-A_1^*)$. We will prove that $f_1^* = c(A-A_2^*)$ and $f_2^* = c(A-A_1^*)$: Obviously $f_1^* \leq c(A-A_2^*)$, $f_2^* \leq c(A-A_1^*)$. Assume that $f_1^* < c(A-A_2^*)$. Then we can increase $f_1$ in $G_1^*$ by the max-flow min-cut theorem for a single commodity flow [F-1]. Note that $(A - A_2^*) \bigcap A_2^* = \emptyset$ and this flow increasing procedure does not influence the flow of commodity II on the set of arcs $A_2^*$. This is a contradiction to the assumption that $(x^*,y^*)$ gives a maximal complementary flow. So $f_1^* = c(A-A_2^*)$. Similarly $f_2^* = c(A-A_1^*)$. Thus

$$f_1^* + f_2^* = c_1(A - A_2^*) + c_2(A - A_1^*).$$

That

$$c_1(A - A_2^*) + c_2(A - A_1^*) = \max_{A_1,A_2} \{c_1(A - A_2) + c_2(A - A_1)\}$$

is obvious since otherwise there exists another partitioning of $A$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

which generates a complementary flow whose flow value is greater than $f_1^* + f_2^*$ . Q.E.D.

3. Minimal Cost Complementary Flow Problem

Let us assume that the feasibility of the complementary flow has been established by solving problem 1. Our problem here is to:

$$(3.2.8) \qquad \begin{cases} \text{minimize} \quad \varphi_o(x,y) = (p^1)^t x + (p^2)^t y \\ \text{subject to} \quad x \in X \ , \ y \in Y \\ \qquad\qquad\qquad x^t y = 0 \end{cases}$$

Theorem 3.2.2. If $b^1$, $b^2$, $p^1$, $p^2$ and $c$ are all vectors of integer components, then (3.2.8) is equivalent to the following bilinear programming problem

$$(3.2.9) \qquad \begin{cases} \text{minimize} \quad \varphi_M(x,y) = (p^1)^t x + (p^2)^t y + Mx^t y \\ \text{subject to} \quad x \in X \ , \ y \in Y \end{cases}$$

provided $M > M_1 - M_2$ where

$$(3.2.10) \qquad \begin{cases} M_1 = \min \{\varphi_o(x,y) \mid x \in X \ , \ y \in Y \ , \ x^t y = 0\} \\ M_2 = \min \{\varphi_o(x,y) \mid x \in X \ , \ y \in Y\} \end{cases}$$

Proof. Let $(x^*, y^*)$ be an extreme optimal solution (which always exists by Theorem 2.1.1 of Part I) of (1.2.9). Then $\varphi_M(x^*, y^*) \leq \varphi_M(x,y)$ for all $x \in X$ and all $y \in Y$ . In particular,

$$\begin{aligned} \varphi_M(x^*, y^*) &\leq \min \{\varphi_M(x,y) \mid x \in X \ , \ y \in Y \ , \ x^t y = 0\} \\ &= \min \{\varphi_o(x,y) \mid x \in X \ , \ y \in Y \ , \ x^t y = 0\} = M_1 \ . \end{aligned}$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Hence

$$M(x^*)^t y^* \leq M_1 - \varphi_o(x^*,y^*)$$
$$\leq M_1 - \min\{\varphi_o(x,y) \mid x \in X,\ y \in Y\}$$
$$= M_1 - M_2$$

It follows from this that $(x^*)^t y^* = 0$ : By assumption $(x^*,y^*)$ is an extreme optimal solution which implies that both $x^*$ and $y^*$ are integer vectors. So if $(x^*)^t y^* \neq 0$, then $(x^*)^t y^* \geq 1$. Hence $(x^*)^t y^* \neq 0$ implies $M(x^*)^t y^* \geq M > M_1 - M_2$, which is a direct contradiction to the inequality proved above. So $(x^*)^t y^* = 0$ has been established. Now we have $\varphi_M(x^*,y^*) = \varphi_o(x^*,y^*)$, so that $\varphi_o(x^*,y^*) \leq \varphi_M(x,y)$ for all $x \in X$ and $y \in Y$. In particular, $\varphi_o(x^*,y^*) \leq \varphi_o(x,y)$ for all $x \in X$, $y \in Y$ satisfying $x^t y = 0$. Thus

$$\varphi_o(x^*,y^*) \leq \min\{\varphi_o(x,y) \mid x \in X,\ y \in Y,\ x^t y = 0\}$$

Since $x^* \in X$, $y^* \in Y$ and $(x^*)^t y^* = 0$, we have

$$\varphi_o(x^*,y^*) \geq \min\{\varphi_o(x,y) \mid x \in X,\ y \in Y,\ x^t y = 0\}$$

so

$$\varphi_o(x^*,y^*) = \min\{\varphi_o(x,y) \mid x \in X,\ y \in Y,\ x^t y = 0\}$$

i.e., $(x^*,y^*)$ is an optimal solution of (3.2.8). Q.E.D.

The exact determination of $M_1$ or $M_2$ is not an easy task. But these constants can be replaced by $M_1'$ and $M_2'$ such that $M_1' \geq M_1$ and $M_2' \leq M_2$. Let $(x^o,y^o)$ be any solution satisfying $x^o \in X$, $y^o \in Y$ and $(x^o)^t y^o = 0$ and let

$$M_1' = \varphi_o(x^o,y^o)$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Then

$$M_1' \geq \min \{\varphi_o(x,y) \mid x \in X\ ,\ y \in Y\ ,\ x^t y = 0\} = M_1$$

Let $q_i = \min\{p_i^1, p_i^2\}$ and let $q$ be the vector having $q_i$ as its $i^{th}$ component. Let $\varphi_o'(x,y) = q^t x + q^t y$ . Then $\varphi_o'(x,y) \leq \varphi(x,y)$ for all $x \geq 0$ , $y \geq 0$ . So

$$\begin{aligned} M_2' &= \min \{\varphi_o'(x,y) \mid x \in X\ ,\ y \in Y\} \\ &\leq \min \{\varphi_o(x,y) \mid x \in X\ ,\ y \in Y\} = M_2 \end{aligned}$$

Note that $M_1'$ is obtained by solving the feasibility problem and that $M_2'$ is obtained by solving a standard minimal cost flow problem with two sources and two sinks. Both of these problems are much easier than (3.2.8) itself and a very efficient algorithm exists for the subproblem associated with $M_2'$ .

In passing, let us briefly discuss the obvious extension of the two commodity complementary flow problem to the general $k$ commodity case. Let $x^i$ represent the vector of flow of the $i^{th}$ commodity $i=1,\ldots,k$ . As before, they have to satisfy the constraints

$$(3.2.11) \qquad \begin{cases} Px^i = b^i\ , \\ 0 \leq x^i \leq c \end{cases} \qquad i=1,\ldots,k$$

where $A, b^i, c$ are defined analogously to the case of $k = 2$ . We will only state the extended bilinear programming formulation of the associated complementary flow problem in the following.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

a. Feasibility Problem

$$(3.2.12)\qquad \begin{cases} \text{minimize} & \displaystyle\sum_{i=1}^{k}\sum_{j=i+1}^{k} (x^i)^t x^j \\ \text{subject to} & Px^i = b^i \\ & 0 \le x^i \le c \end{cases} \qquad i=1,\ldots,k$$

b. Minimum Cost Complementary Flow Problem

$$(3.2.13)\qquad \begin{cases} \text{minimize} & \displaystyle\sum_{i=1}^{k} (p^i)^t x^i + M \sum_{i=1}^{k}\sum_{j=i+1}^{k} (x^i)^t x^j \\ \text{subject to} & Px^i = b^i \\ & 0 \le x^i \le c \end{cases} \qquad i=1,\ldots,k$$

Both of these problems can possibly be solved by the extended bilinear programming algorithm developed in Part I, Chapter 5.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

## 3.3 Orthogonal Production Schedules

Let us consider the optimization of the following multi-stage production system:

$$(3.3.1)\qquad \begin{cases} \text{minimize} & \displaystyle\sum_{\ell=1}^{L}\sum_{j=1}^{n} c_j^{\ell} x_j^{\ell} & \\ \text{subject to} & \displaystyle\sum_{j=1}^{n} a_{ij} x_j^{\ell} \geq b_i^{\ell} & i=1,\ldots,m \\ & & \ell=1,\ldots,L \\ & x_j^{\ell} \geq 0\ ,\ j=1,\ldots,n\ ; & \ell=1,\ldots,L \end{cases}$$

where

$b_i^{\ell}$ = demand for commodity $i$ at period $\ell$

$c_j^{\ell}$ = unit cost associated with activity $j$ at period $\ell$

$a_{ij}$ = input - output coefficients

$x_j^{\ell}$ = level of activity $j$ at period $\ell$ .

If there exist no other constraints, then (3.3.1) is separable into $L$ independent linear programs, but if we add the following constraints

$$(3.3.2)\qquad x_j^{\ell-1} \cdot x_j^{\ell} = 0 \qquad j \in J\ , \qquad \ell=2,\ldots,L$$

for certain index set $J$ , then the system (3.3.1), (3.3.2) is no longer a linear programming problem. The physical interpretation of these constraints will be

(a) due to the maintenance of machines, certain activities cannot be used in two consecutive periods (machine scheduling problem);



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

(b) due to the hysterisis effect, very high cost is incurred if we use certain activity in two consecutive periods (agricultural scheduling problems).

The set of vectors $(x^1, \ldots, x^L)$ satisfying the constraints of (3.3.1) and (3.3.2) will be called an L period J-orthogonal production schedule. Using the vector matrix notation, we have the system

$$(3.3.3) \qquad \begin{cases} \text{minimize} \quad f(x^1, \ldots, x^L) = \sum_{\ell=1}^{L} (c^\ell)^t x^\ell \\ \text{subject to} \quad Ax^\ell \geq b^\ell \,, \\ \qquad\qquad x^\ell \geq 0 \qquad\qquad \ell = 1, \ldots, L \\ \qquad\qquad x_j^{\ell-1} \cdot x_j^\ell = 0 \quad ; \quad j \in J \end{cases}$$

As in the previous section, the following problems naturally arise

a. Feasibility Problem. Does there exist a feasible J orthogonal schedule?

b. Minimum Cost J-Orthogonal Production Scheduling Problem. What is the minimum cost J-orthogonal production schedule among all the J-orthogonal production schedules?

As before, the feasibility problem can be formulated as the following bilinear programming problem

$$(3.3.4) \qquad \begin{cases} \text{minimize} \quad \varphi(x^1, \ldots, x^L) = \sum_{\ell=2}^{L} \sum_{j \in J} x_j^{\ell-1} x_j^\ell \\ \text{subject to} \quad Ax^\ell \geq b^\ell \\ \qquad\qquad x^\ell \geq 0 \end{cases} \qquad \ell = 1, \ldots, L$$

while the minimum cost J-orthogonal production scheduling problem is



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

reducible to the following bilinear programming problem for large enough $M$

$$(3.3.5)\quad \begin{cases} \text{minimize} \quad \varphi(x^1,\ldots,x^L) = \sum (c^\ell)^t x^\ell + M \sum_{\ell=2}^{L} \sum_{j\in J} x_j^{\ell-1} x_j^\ell \\ \\ \text{subject to} \quad Ax^\ell \geq b^\ell , \qquad \ell=1,\ldots,L \\ \qquad\qquad\quad x^\ell \geq 0 \quad . \end{cases}$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

# 4. REDUCTION OF A CERTAIN CLASS OF MATHEMATICAL PROGRAMMING PROBLEMS TO EQUIVALENT BILINEAR PROGRAMMING PROBLEMS

## 4.1 Maximization of a Convex Quadratic Function over a Polyhedral Convex Set

Let us consider a quadratic programming problem

$$(4.1.1) \qquad \begin{cases} \underset{z}{\text{maximize}} & f(z) = 2c^t z + z^t Q z \\ \text{subject to} & Az \leq b \\ & z \geq 0 \end{cases}$$

where $z \in R^n$, $c \in R^n$, $A \in R^{m \times n}$, $b \in R^m$ and $Q \in R^{n \times n}$ is a symmetric positive semi-definite matrix and the constraint set

$$(4.1.2) \qquad Z = \{z \mid Az \leq b \ , \ z \geq 0\} \neq \emptyset$$

is bounded. Obviously there exists an optimal solution $z^*$ which is an extreme point of $Z$ since $f$ is convex and $Z$ is bounded.

0-1 integer programming problems, fixed charge problems etc., can be reformulated as this class of problems, but only a few algorithms have been proposed despite its importance. In fact, if we exclude the total enumeration of all the extreme points of $Z$, then Hitter's cutting plane algorithm [R-2] and Tui's algorithm [T-3] are the only proposed algorithms to solve this class of problems. Unfortunately, however, the finite convergence of these algorithms have not been established.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Here we will consider the reduction of (4.1.1) into an equivalent bilinear programming problem and compare our BLP algorithm with other algorithms.

Theorem 4.1.1. (Altman [A-1]). Let $(x^*,y^*)$ be an optimal solution of the following BLP:

$$(4.1.3)\qquad \begin{cases} \text{maximize} & \varphi(x,y) = c^t x + c^t y + x^t Q y \\ \text{subject to} & Ax \leq b\ ,\ x \geq 0 \\ & Ay \leq b\ ,\ y \geq 0 \end{cases}$$

Then both $x^*$ and $y^*$ solve (4.1.1). In addition, if $Q$ is positive definite, then $x^* = y^*$.

Proof. Let $z^*$ be an optimal solution of (4.1.1). Then by definition $f(z^*) \geq f(z)$ for all $z \in Z$. In particular, $f(z^*) \geq f(x^*) = \varphi(x^*,x^*)$ and $f(z^*) \geq f(y^*) = \varphi(y^*,y^*)$. In addition, we have $f(z^*) \leq \varphi(x^*,y^*)$ since

$$\begin{aligned} \varphi(x^*,y^*) &= \max\{\varphi(x,y) \mid x \in Z\ ,\ y \in Z\} \\ &\geq \max\{\varphi(z,z) \mid z \in Z\} = f(z^*) \end{aligned}$$

Now it suffices to show that $\varphi(x^*,y^*) = \varphi(x^*,x^*) = \varphi(y^*,y^*)$ since this will establish $f(z^*) = \varphi(x^*,x^*) = \varphi(y^*,y^*)$ by the facts proved above. By the optimality of $(x^*,y^*)$ relative to (4.1.3), we have $\varphi(x^*,y^*) \geq \varphi(x^*,x^*)$ and $\varphi(x^*,y^*) \geq \varphi(y^*,y^*)$. Hence

$$\varphi(x^*,y^*) - \varphi(x^*,x^*) = c^t(y^* - x^*) + (x^*)^t Q(y^* - x^*) \geq 0$$
$$\varphi(x^*,y^*) - \varphi(y^*,y^*) = c^t(x^* - y^*) + (y^*)^t Q(x^* - y^*) \geq 0$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Adding these two inequalities, we have $(x^* - y^*)^t Q(x^* - y^*) \leq 0$. By the positive semi-definiteness of $Q$, we have $Q(x^* - y^*) = 0$ (if $Q$ is positive definite, then $x^* = y^*$ follows.) Putting this into the inequalities above, we see $c^t(y^* - x^*) \geq 0$ and $c^t(x^* - y^*) \geq 0$. So $c^t(x^* - y^*) = 0$. Hence $\varphi(x^*, y^*) - \varphi(x^*, x^*) = 0$ and $\varphi(x^*, y^*) - \varphi(y^*, y^*) = 0$ as desired. Q.E.D.

The BLP (4.1.3) has twice as many constraints and variables as its counterpart (4.1.1) and it might look less favorable to solve (4.1.3) by BLP algorithm than to apply Ritter's cutting plane algorithm or Tui's algorithm directly to (4.1.1). But our algorithm is more powerful than the others under certain conditions since

(i) it guarantees the finite convergence to an extreme ε-optimal solution;

(ii) it suffices to maintain a single simplex tableau instead of maintaining two simplex tableaux corresponding to $x$ and $y$;

(iii) it sometimes generates a deeper cut at a locally maximum vertex.

Statement (i) has been established in Part I, so we will discuss only (ii) and (iii).

4.1.1 Modification of the Suboptimization Phase (Step 2 of BLP Algorithm)

First, let us prove the following proposition.

Proposition 4.1.2. If $Q$ is positive semi-definite, then $\varphi(x,y) \geq \varphi(y,y)$ implies $\varphi(x,x) \geq \varphi(x,y)$. Moreover $\varphi(x,y) > \varphi(y,y)$ implies $\varphi(x,x) > \varphi(x,y)$.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Proof. Obviously $\varphi(x,y) \geq (>)\ \varphi(y,y)$ if, and only if $(x-y)^t(c + Qy) \geq (>)\ 0$

$$\begin{aligned}\varphi(x,x) - \varphi(x,y) &= c^t(x - y) + x^tQ(x - y)\\ &= c^t(x - y) + y^tQ(x - y) + (x - y)^t\, Q(x - y)\\ &\geq (c^t + y^tQ)(x - y) = (x - y)^t(c + Qy)\end{aligned}$$

Hence $\varphi(x,y) \geq (>)\ \varphi(y,y)$ implies $\varphi(x,x) \geq (>)\ \varphi(x,y)$ . Q.E.D.

Based upon this proposition, we will modify the algorithms as follows: Given $x^j$ , define $x^{j+1}$ by the formula

$$x^{j+1} = \arg\max \{\varphi(x,x^j) \mid x \in Z\} .$$

Obviously $\varphi(x^{j+1},x^j) \geq \varphi(x^j,x^j)$ . By Proposition 4.1.2, $\varphi(x^{j+1},x^{j+1}) \geq \varphi(x^{j+1},x^j)$ . So we will eliminate the computation of $y^{j+1}$ and simply set $y^{j+1} = x^{j+1}$ . This enables us to totally neglect the simplex tableau corresponding to $y$ . As in the original BLP algorithm, this process terminates under the condition $x^{k+1} = x^k \stackrel{d}{=} x^\infty$ in finitely many steps. It is easy to see that this process corresponds to the steepest ascent along the edges relative to the coefficients of the linear terms of the objective function. The rest of the sub-optimization procedure remains the same and it will produce a locally maximum vertex $(x^p,y^p)$ . Of course, $x^p$ is a locally maximum vertex relative to (4.1.1).

4.1.2 Comparison of the Various Cuts

Let us assume that (4.1.1) and (4.1.3) have been transformed into their canonical forms relative to $x^p$ and $(x^p,y^p)$ , respectively, i.e., $c \leq 0$ and $b \geq 0$ . We will assume for simplicity that $c < 0$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

and $b > 0$ since the comparison of various cuts is much easier for this case than for the degenerate case.

(a) Ritter's Cut

Let

$$\psi_R(\sigma) = \max \{f(z) \mid -c^t z \leq \sigma,\ z \geq 0\}$$

then Ritter's cut is defined by $-c^t z \geq \sigma_R$ where

$$\sigma_R = \max \{\tau \mid \psi_R(\sigma) \leq \varphi_o - \varphi_R,\ \sigma \in [0,\tau]\}$$

That this cut does not eliminate any point $z \geq 0$ such that $f(z) > \varphi_o - \varphi_R$ follows directly from the definition of $\sigma_R$. By the elementary calculation, the explicit expression for $\sigma_R$ is obtained as follows:

$$\sigma_R = \frac{1 + \sqrt{1 + (\varphi_o - \varphi_p)\bar{q}}}{\bar{q}}$$

where

$$\bar{q} = \max \left\{ \frac{q_{ii}}{c_i^2} \;\middle|\; i = 1, \ldots, n \right\}$$

$q_{ii} \geq 0$ for all $i$ since $Q$ is positive semi-definite and $c_i < 0$ for all $i$ by assumption, so that $\sigma_R > 0$.

(b) Tui's Cut

Let $z_i^*$ be the greatest root of the equation

$$2c_i z + q_{ii} z^2 = \varphi_o - \varphi_p$$

i.e.,

$$z_i^* = \frac{1 + \sqrt{1 + (\varphi_o - \varphi_p)\bar{q}_{ii}}}{-c_i \bar{q}_{ii}}$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

where $\bar{q}_{ii} = q_{ii}/c_i^2$ . Obviously $z_i^* > 0$ for all $i$ . Then Tui's cut is defined by

$$\sum_{i=1}^{n} z_i/z_i^* \geq 1$$

It follows from the convexity of $f$ and the definition of $z_i^*$ that this cut is 'legitimate'.

(c) BLP Cuts

Let

$$\psi_B(\sigma) = \max \{\varphi(x,y) \mid -c^t x \leq \sigma \ , \ x \geq 0 \ , \ y \in Z\}$$

and let

$$\sigma_B = \max \{\tau \mid \psi_B(\sigma) \leq \varphi_o - \varphi_p \ , \ \sigma \in [0,\tau]\} \ .$$

Theorem 4.1.3. The cut $-c^t x \geq \sigma_B$ does not eliminate any point $x \in Z$ such that $f(x) > \varphi_o - \varphi_p$ .

Proof. By definition, $\varphi(x,y) \leq \varphi_o - \varphi_p$ for all $y \in Z$ and all $x \in Z \cap S(\sigma_B)$ where $S(\sigma_B) = \{s \in R^n \mid -c^t s \leq \sigma_B, s \geq 0\}$ Hence $\varphi(x,y) \leq \varphi_o - \varphi_p$ for all $x \in Z \cap S(\sigma_B)$ and $y \in Z \cap S(\sigma_B)$ . So $f(x) = \varphi(x,x) \leq \varphi_o - \varphi_p$ for all $x \in Z \cap S(\sigma_B)$ . Q.E.D.

Example 1. Let us consider the problem

$$\begin{cases} \text{maximize} & f(z) = -2z_1 - 3z_2 + 2z_1^2 - 2z_1z_2 + 2z_2^2 \\ \text{subject to} & (z_1,z_2) \in Z \end{cases}$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

where Z is illustrated in Fig. 4.1.1 and assume that we have $\varphi_o = \varphi(3,3) = 3$ .

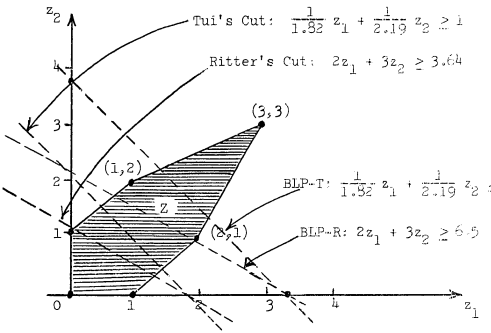



Figure 4.1.1.

(a) Ritter's Cut

$$\psi_R(\sigma) = \max\{-2z_1 - 3z_2 + 2z_1^2 - 2z_1z_2 + 2z_2^2 \mid 2z_1 + 3z_2 \leq \sigma,\ z_1 \geq 0,\ z_2 \geq 0\}$$
$$= \max\{0,\ -\sigma + \tfrac{1}{2}\sigma^2,\ -\sigma + \tfrac{2}{9}\sigma^2\}$$

So $\sigma_R$ is given by the positive root of the equation $-\sigma + \frac{1}{2}\sigma^2 = 3$ , namely $\sigma_R = 1 + \sqrt{7} \doteq 3.64$ and Ritter's cut is given by

$$2z_1 + 3z_2 \geq 3.64 \quad .$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

(b) Tui's Cut

$z_1^*$ and $z_2^*$ are the positive roots of the equations $-2z_1 + 2z_1^2 = 3$ and $-3z_2 + 2z_2^2 = 3$, respectively. Hence $z_1^* = (1 + \sqrt{7})/2 \doteq 1.82$, $z_2^* = (3 + \sqrt{33})/4 \doteq 2.19$ and Tui's cut given by

$$\frac{1}{1.82} z_1 + \frac{1}{2.19} z_2 \geq 1 \quad .$$

(c) BLP Cut. $g_1 x_1 + g_2 x_2 \geq \sigma$.

$$\varphi(x,y) = -x_1 - \frac{3}{2} x_2 - y_1 - \frac{3}{2} y_2 + 2x_1 y_1 - x_1 y_2 - x_2 y_1 + 2x_2 y_2$$

$$\psi(\sigma) = \max\{\varphi(x,y) \mid g_1 x_1 + g_2 x_2 \leq \sigma,\ x_1 \geq 0,\ x_2 \geq 0,\ y \in Z\}$$

$$= \max\{0\ ,\ \psi_{B_1}(\sigma)\ ,\ \psi_{B_2}(\sigma)\}$$

where

$$\psi_{B_1}(\sigma) = -\sigma/g_1 + \max\{-y_1 - \frac{3}{2} y_2 + \frac{\sigma}{g_1}(2y_1 - y_2) \mid y \in Z\}$$

$$\psi_{B_2}(\sigma) = -\frac{3\sigma}{2g_2} \max\{-y_1 - \frac{3}{2} y_2 + \frac{\sigma}{g_2}(-y_1 + 2y_2) \mid y \in Z\}$$

It is not difficult to see that

$$\psi_{B_1}(\sigma) = \begin{cases} -\sigma/g_1 & 0 \leq \sigma/g_1 \leq 1/2 \\ \sigma/g_1 - 1 & 1/2 \leq \sigma/g_1 \leq 5/2 \\ 2\sigma/g_1 - 7/2 & \sigma/g_1 \geq 5/2 \end{cases}$$

$$\psi_{B_2}(\sigma) = \begin{cases} -3\sigma/2g_2 & 0 \leq 3\sigma/2g_2 \leq 9/8 \\ \sigma/2g_2 - 3/2 & 9/8 \leq 3\sigma/2g_2 \leq 15/4 \\ 3\sigma/2g_2 - 4 & 3\sigma/2g_2 \geq 15/4 \end{cases} \quad .$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

(1) BLP-R Cut: $g_1 = 1$ , $g_2 = 3/2$

$$\psi_{B_1}(\sigma) = 3 \quad \text{gives} \quad \sigma_{B_1} = 13/4 = 3.25$$

$$\psi_{B_2}(\sigma) = 3 \quad \text{gives} \quad \sigma_{B_2} = 7 \ .$$

BLP-R: $z_1 + \frac{3}{2} z_2 \geq 3.25$

(2) BLP-T Cut: $g_1 = 1/1.82$ , $g_2 = 1/2.19$

$$\psi_{B_1}(\sigma) = 3 \quad \text{gives} \quad \sigma_{B_1} = 3.25/1.82 = 1.78$$

$$\psi_{B_2}(\sigma) = 3 \quad \text{gives} \quad \sigma_{B_2} = 7/2.19 = 3.18 \ .$$

BLP-T: $\frac{1}{1.82} z_1 + \frac{1}{2.19} z_2 \geq 1.78$ .

All four cuts are illustrated in Fig. 4.1.1. It is seen from this that the strengths are of the following order: (i) BLP-T cut; (ii) BLP-R cut; (iii) Tui's cut; (iv) Ritter's cut for this particular example.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Example 2. Let us consider the problem

$$\begin{cases} \text{maximize} & f(z) = -2z_1 - 3z_2 + 2z_1^2 + 2z_2^2 \\ \text{subject to} & (z_1, z_2) \in Z \end{cases}$$

where $Z$ is the same set as before. Let us assume that $\varphi_o = \varphi(0,0) = 0$. Similar calculation shows that

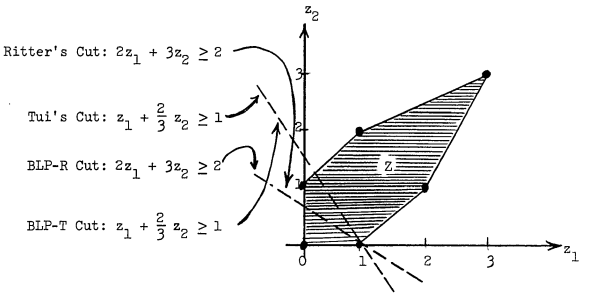



Figure 4.1.2

It can be seen from this that Tui's cut and BLP-T cut are identical and Ritter's cut and BLP-R cut coincides with each other. The former two are stronger than the latter two.

It will be observed from these two examples that Tui's cut is stronger than Ritter's cut and it can be proved that this is always the case (see expressions in the definitions of these cuts). However, we cannot give a general statement about the relative strength of Tui's cut and BLP cuts. It depends heavily on the shape of the feasible



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

region and the properties of the objective function. But the following observation will be worthwhile for deciding whether or not we should consider the introduction of a BLP cut.

(i) If $Q$ contains many negative off diagonal entries, then it will work in favor of the BLP cut. This is more so if their absolute values are bigger.

(ii) If $\varphi_o - \varphi_p$ is large, then BLP cuts tend to be stronger than Tui's cut since $\sigma_B$ is a root of a <u>linear</u> equation while $\sigma_T$ is a root of a <u>quadratic</u> equation.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

## 4.2 0-1 Integer Programs

Let us consider the 0-1 integer program in inequality form defined below:

$$(4.2.1)\qquad \begin{cases} \text{maximize} & c^t x \\ \text{subject to} & Ax \le b \\ & x_j = 0 \text{ or } 1\ , \qquad j=1,\ldots,n \end{cases}$$

where $x = (x_1,\ldots,x_n)^t$ , $c \in R^n$ , $A \in R^{m\times n}$ and $b \in R^m$ .

It is well known that any integer program can be reduced to the 0-1 integer program and there are numerous references to this problem ([B-1], [G-3], for example). Here we will propose another approach to (4.2.1) via BLP algorithm.

### 4.2.1 A Feasible Solution of a 0-1 Integer Program

Let $X$ be the set of the feasible solutions of (4.2.1), i.e.,

$$(4.2.2)\qquad X = \{x \in R^n \mid Ax \le b\ ,\ x_j = 0 \text{ or } 1\ ,\ j=1,\ldots,n\}$$

It is well known that the problem of finding a feasible solution $x \in X$ is equivalent to the problem of finding a vector $x$ satisfying

$$(4.2.3)\qquad \begin{cases} Ax \le b \\ 0 \le x \le e_n \\ x^t(e_n - x) = 0 \end{cases}$$

where $e_n$ is the n-dimensional vector of 1's. The equivalence follows since $x_j = 0$ or $1$ for all $j$ if $0 \le x \le e_n$ and $x^t(e_n - x) = 0$ .



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Obviously, this problem is reducible to the following quadratic programming problem ([R-5])

$$(4.2.4) \qquad \begin{cases} \text{maximize} & f(x) = x^t(x - e_n) \\ \text{subject to} & Ax \leq b \\ & 0 \leq x \leq e_n \end{cases}$$

Let $x^*$ be an optimal solution of (4.2.4). It is easy to see that if $f(x^*) = 0$, then $x^*$ belongs to $X$ while if $f(x^*) < 0$, then $X = \emptyset$.

Let us introduce the BLP associated with (4.2.4):

$$(4.2.5) \qquad \begin{cases} \text{maximize} & \varphi(x,y) = -\frac{1}{2} e_n^t x - \frac{1}{2} e_n^t y + x^t I y \\ \text{subject to} & x \in X_o \,,\ y \in X_o \end{cases}$$

where $I$ is the $n \times n$ identity matrix and

$$(4.2.6) \qquad X_o = \{x \in R^n \mid Ax \leq b \,,\ 0 \leq x \leq e_n\}$$

Theorem 4.2.1. Let $(x^*,y^*)$ be an optimal solution of (4.2.5). If $\varphi(x^*,y^*) = 0$, then both $x^*$ and $y^*$ belong to $X$ while if $\varphi(x^*,y^*) < 0$, then $X = \emptyset$. Moreover, $x^* = y^*$.

Proof†. By Theorem 4.1.1, both $x^*$ and $y^*$ are optimal solutions of (4.2.4). Obviously $I$ is positive definite and the result follows.

Q.E.D.

†G. Dantzig [D-4] has given an alternative proof of the simpler version of this theorem without recourse to Theorem 4.1.1. He also succeeded in reducing the problem of finding a vector $x \in X$ to the standard linear complementarity problem.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

When we apply the BLP algorithm to (4.2.5) to find a feasible solution of (4.2.1), we can always replace $\varphi_\ell$ by 0, so that the cuts generated at each non-optimal locally maximum vertex is expected to be very deep. (The discussion about the relative strength of Tui's cut and BLP cuts apply here.) Also note that the structure of the constraint set $X_o$ enables us to apply the efficient upper bounding technique [D-1] in the suboptimization step. Therefore, the BLP algorithm is expected to generate an $\epsilon$-optimal solution very efficiently.

Theorem 4.2.2. Let $X \neq \emptyset$ and let $(\bar{x},\bar{y})$ be an $\epsilon$-optimal solution of (4.2.5), i.e., $\varphi(\bar{x},\bar{y}) \geq -\epsilon$, where $\epsilon$ is a small positive constant. Then $\bar{x}_j \leq 2\epsilon$ or $\bar{x}_j \geq 1 - 2\epsilon$ for all $j$. Also either $\bar{y}_j \leq 2\epsilon$ or $\bar{y}_j \geq 1 - 2\epsilon$ for all $j$.

Proof. The following inequality holds by assumption

$$\bar{x}^t(e_n - \bar{y}) + \bar{y}^t(e_n - \bar{x}) \leq 2\epsilon \tag{4.2.7}$$

Since $\bar{x}_j(1 - \bar{y}_j) \geq 0$ and $\bar{y}_j(1 - \bar{x}_j) \geq 0$ for all $j$, we have

$$\bar{x}_j(1 - \bar{y}_j) + \bar{y}_j(1 - \bar{x}_j) \leq 2\epsilon, \quad j=1,\ldots,n.$$

Suppose $2\epsilon < \bar{x}_j < 1-2\epsilon$ for some $j$. Then

$$\bar{x}_j(1 - \bar{y}_j) + \bar{y}_j(1 - \bar{x}_j) > 2\epsilon(1 - \bar{y}_j) + \bar{y}_j \cdot 2\epsilon = 2\epsilon$$

which is a contradiction to the inequality (4.2.7). Hence either $\bar{x}_j \leq 2\epsilon$ or $\bar{x}_j \geq 1-2\epsilon$. The statement for $\bar{y}_j$ can be proved analogously. Q.E.D.

If $\epsilon$ is small enough, then by truncating $\bar{x}_j$ and $\bar{y}_j$ down to 0 or up to 1, we can obtain a point $x \in X$ if it does not violate the constraint $Ax \leq b$.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

4.2.2 An Optimal Solution of a 0-1 Integer Program

Let us assume $X \neq \emptyset$ and turn now to the original problem (4.2.1), which is equivalent to the problem:

$$(4.2.8) \qquad \begin{cases} \text{maximize} & c^t x \\ \text{subject to} & x \in X_o \\ & x^t(e_n - x) = 0 \end{cases}$$

We will introduce the QP associated with (4.2.8):

$$(4.2.9) \qquad \begin{cases} \text{maximize} & f_M(x) = c^t x - Mx^t(e_n - x) \\ \text{subject to} & x \in X_o \end{cases}$$

where $M$ is a positive constant.

Theorem 4.2.3. If $M > M_o \equiv \sum_{j=1}^{n} |c_j|/\delta$, then the optimal solution $x^*$ of (4.2.9) satisfies $(x^*)^t(e_n - x^*) \leq \delta$. Moreover, if $(x^*)^t(e_n - x^*) = 0$, then $x^*$ gives an optimal solution of (4.2.8).

Proof. By the optimality of $x^*$ relative to (4.2.9), we have

$$(4.2.10) \quad c^t x^* - M(x^*)^t(e_n - x^*) \geq c^t x - Mx^t(e_n - x) \qquad \forall\, x \in X_o$$

Let $x'$ be a feasible solution of (4.2.1). Obviously $x' \in X_o$ and $(x')^t(e_n - x') = 0$. It follows that

$$M(x^*)^t(e_n - x^*) \leq c^t x^* - c^t x' \leq \sum_{j=1}^{n} |c_j|$$

Hence if $M > M_o \equiv \sum_{j=1}^{n} |c_j|/\delta$, then

$$(x^*)^t(e_n - x^*) \leq \sum_{j=1}^{n} |c_j|/M < \delta .$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Moreover, if $(x^*)^t(e_n - x^*) = 0$ , then

$$c^t x^* \geq c^t x \qquad \forall\, x \in X$$

by the inequality (4.2.9) and $x^*$ is an optimal solution of (4.2.1).

Q.E.D.

As before, $f_M(x)$ is a convex function for all $M \geq 0$ and we will introduce the BLP associated with (4.2.9):

$$(4.2.11)\begin{cases}\text{maximize } \varphi_M(x,y) = \frac{1}{2} c^t x + \frac{1}{2} c^t y - M(\frac{1}{2} e_n^t x + \frac{1}{2} e_n^t y - x^t y) \\ \text{subject to} \qquad x \in X_o \,,\ y \in Y_o\end{cases}$$

Corollary 4.2.3.1. If $M > M_o \equiv \Sigma|c_j|/\delta$ , then the optimal solution $(x^*,y^*)$ of (4.2.11) satisfies $x^* = y^*$ and $\varphi(x^*,y^*) \equiv \frac{1}{2} e_n^t x^* + \frac{1}{2} e_n^t y^* - (x^*)^t y^* \leq \delta$ . If, in addition, $\varphi(x^*,y^*) = 0$ , then both $x^*$ and $y^*$ are optimal solutions of (4.2.1).

Proof. Follows directly from Theorem 4.1.1 and 4.2.3. Q.E.D.

From the computational point of view, the existence of large $M$ in the objective function will cause serious trouble and the efficiency of the BLP algorithm applied to (4.2.11) is not clear.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

# 5. MISCELLANEOUS BILINEAR PROGRAMMING PROBLEMS

## 5.1 Reduction of a Sparse Matrix into an Almost-Triangular Matrix

### 5.1.1 Reduction by Arbitrary Row and Column Permutations

Let $A = (a_{ij})$ be a given $n \times n$ matrix which contains many zero entries. It is sometimes desirable to permute the rows and columns of this sparse matrix and rearrange it into an almost-triangular matrix. For example, if a matrix $A$ with apparently no structure has been transformed into an almost-triangular matrix, then (i) the storage space necessary to store the entries of $A$ will be reduced to almost the half of (randomly distributed) original matrix and, (ii) the Gaussian elimination to calculate $A^{-1}$ will be greatly simplified by using its almost triangular structure (see [W-2]).

Definition 5.1.1. A square $n \times n$ matrix $D = (d_{ij})$ will be called $\ell$-upper (lower)-triangular if it contains $\ell$ nonzero entries $d_{ij}$ for $i > j$ $(i < j)$. $\ell$ will be called an index of non-upper (lower)-triangularity of $D$. Of course, $D$ is upper (lower)-triangular if its index of non-upper (lower)-triangularity is zero.

Let $B = (b_{ij})$ be a $n \times n$ binary matrix associated with $A$:

$$b_{ij} = \begin{cases} 1 & \text{if } a_{ij} \neq 0 \\ 0 & \text{if } a_{ij} = 0 \end{cases} \tag{5.1.1}$$

and let $\mathcal{P}_n$ be the class of $n \times n$ permutation matrices, i.e.,



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$\mathcal{P}_n = P = \left\{ (p_{ij}) \;\middle|\; \sum_{j=1}^{n} p_{ij} = 1 \ , \ i=1,\ldots,n \ ; \ \sum_{i=1}^{n} p_{ij} = 1, \quad j=1,\ldots,n \ ; \right.$$

$$\left. p_{ij} = 0 \ \text{ or } \ 1 \ , \ i,j=1,\ldots,n \right\} \tag{5.1.2}$$

Let $X = (x_{ij}) \in \mathcal{P}_n$ and $Y = (y_{ij}) \in \mathcal{P}_n$ and define $B^{X,Y} = (b_{ij}^{X,Y})$ by

$$B^{X,Y} = Y \, B \, X \tag{5.1.3}$$

Then the non-upper triangularity index $\ell(X,Y)$ of $B^{X,Y}$ is given by

$$\begin{aligned} \ell(X,Y) &\equiv \sum_{i=1}^{n} \sum_{j=1}^{i-1} b_{ij}^{X,Y} \\ &= \sum_{i=1}^{n} \sum_{j=1}^{i-1} \sum_{r=1}^{n} \sum_{s=1}^{n} y_{ir} \, b_{rs} \, x_{sj} \end{aligned} \tag{5.1.4}$$

Our problem is to minimize $\ell(X,Y)$ subject to $X \in \mathcal{P}_n$ , $Y \in \mathcal{P}_n$ .
This is a special case of (2.1.3) for $L = 2$ and is equivalent to the following bilinear programming problem:

$$\left\{ \begin{array}{ll} \text{minimize} & \ell(X,Y) = \displaystyle\sum_{i=1}^{n} \sum_{j=1}^{i-1} \sum_{r=1}^{n} \sum_{s=1}^{n} b_{rs} \, y_{ir} \, x_{sj} \\ \text{subject to} & \displaystyle\sum_{j=1}^{n} x_{ij} = 1 \ , \quad \sum_{j=1}^{n} y_{ij} = 1 \ , \quad i=1,\ldots,n \ ; \\ & \displaystyle\sum_{i=1}^{n} x_{ij} = 1 \ , \quad \sum_{i=1}^{n} y_{ij} = 1 \ , \quad j=1,\ldots,n \ ; \\ & x_{ij} \geq 0 \ , \quad y_{ij} \geq 0 \ , \quad i,j=1,\ldots,n \ . \end{array} \right. \tag{5.1.5}$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Since the objective function $\ell(X,Y)$ takes integral value at each extreme point of the constraint set, we can use $\varphi_o' = \varphi_o - 1$ whenever we construct a cutting plane. So the BLP algorithm can generate an optimal solution in finitely many steps.

5.1.2 Reduction by the Simultaneous Permutation of Rows and Columns -- The Reduction of a Directed Graph into a Rooted Directed Graph by the Elimination of the Minimal Number of Arcs

Let us consider now the more restrictive but more interesting case in which we want to minimize the number of nonzero entries in the lower triangular portion of the given matrix by the simultaneous interchange of $i^{th}$ and $j^{th}$ row and $i^{th}$ and $j^{th}$ column.

Let us construct a directed graph $(N,\mathcal{A})$ associated with the given nXn matrix $A = (a_{ij})$ as follows: Let $N = \{1,\ldots,n\}$ be the set of nodes corresponding to n rows and n columns of A . Let us introduce a directed arc $A_{ij}$ from node i to node j if, and only if $a_{ij} \neq 0$ and define $\mathcal{A} = \{A_{ij}\}$ .

Example

$$A = \begin{bmatrix} 0 & X & X & 0 & 0 & 0 \\ 0 & 0 & X & 0 & 0 & 0 \\ 0 & 0 & 0 & X & X & 0 \\ 0 & X & 0 & 0 & 0 & X \\ 0 & X & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & X & 0 \end{bmatrix}$$

X represents the nonzero entry

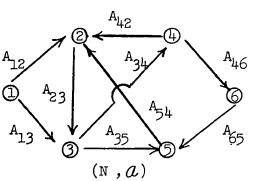



Figure 5.1.1



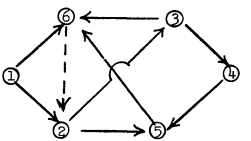
Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Then the problem stated above is equivalent to the problem of reordering the set of n vertices in such a way as to minimize the number of arcs directed from the node with higher index to the node with lower index.

Definition 5.1.2. $(N, \mathcal{A})$ will be called a rooted graph if there is no directed arc connecting nodes i and j such that $i > j$ for an appropriate ordering of n nodes. The associated ordering will be called a consistent ordering of the nodes in a rooted graph.

Using this terminology, our problem is to obtain the minimal set of arcs $\mathcal{A}_0$ such that $(N, \mathcal{A} - \mathcal{A}_0)$ is a rooted graph (always exists since $(N, \mathcal{A} - \mathcal{A})$ is a rooted network) and to obtain the consistent ordering of n nodes relative to $(N, \mathcal{A} - \mathcal{A}_0)$ .

Example. Let $(N, \mathcal{A})$ be a graph given by Fig. 5.1.1. Obviously this is not a rooted graph. But $(N, \mathcal{A} - A_{23})$ is a rooted graph, so that $\mathcal{A}_0 = (A_{23})$. Figure 5.1.2 shows the consistent ordering relative to $(N, \mathcal{A} - A_{23})$ and its corresponding permuted matrix.




$(N, \mathcal{A} - A_{23})$

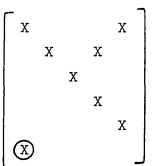



Figure 5.1.2



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Application 1. Ordering of the Activities in a Leontief System

Let us consider a Leontief input-output system of $n$ activities each producing only one product. Let $a_{ij}$ be the amount of product $i$ required to produce a unit amount of product $j$. We will consider the ordering of these $n$ activities based upon their dependence relations. Let us define the relation $\lesssim$ as follows:

$$i \lesssim j \quad \text{if, and only if} \quad a_{ij} \neq 0$$

The transitive law does not necessarily hold for this relation and this does not define an order relation in general. However, if we reorder the activities in such a way as to minimize the number of nonzero entries $a_{ij}$ such that $i > j$, then it will give the most reasonable ordering based upon their mutual dependence.

Application 2. Ranking of $n$ Players in a Contest

Let there be $n$ players in a contest of the game played between two players. The contest is undesigned so that every player need not play the same number of games but each player is not allowed to play against the same opponent more than twice. Given the result of the contest, let us consider the ranking of these $n$ players. One reasonable ranking will be to rank the players in such an order as to minimize the number of games in which the player with lower rank won over the player with higher rank. If we use this as the criterion for the 'best' ranking, then the problem becomes the one stated above.

It follows directly from the discussion in the previous section that this class of problems can be formulated as a quadratic programming



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

problem:

$$(5.1.6)\qquad \begin{cases} \text{minimize} \quad \ell(X) = \sum_{i=1}^{n} \sum_{j=1}^{i-1} \sum_{r=1}^{n} \sum_{s=1}^{n} b_{rs} x_{ir} x_{sj} \\ \text{subject to} \quad X = (x_{ij}) \in \mathcal{P}_n \end{cases}$$

where $B = (b_{rs})$ is the binary matrix associated with the given (sparse) matrix $A = (a_{ij})$ . $\ell(\cdot)$ is neither convex nor concave, but this is reducible to a bilinear programming problem and is solvable by our BLP algorithm as we shall see in the following:

Let

$$(5.1.7)\qquad B(m) = B + m\, I$$

where $m$ is a positive constant and $I$ is an $n \times n$ identity matrix. Let us consider the following problem associated with $B(m)$ :

$$(5.1.8)\qquad \begin{cases} \text{minimize} \quad \ell_m(X,Y) = \sum_{i=1}^{n} \sum_{j=1}^{i-1} \sum_{r=1}^{n} \sum_{s=1}^{n} b_{rs}(m)\, y_{ir} x_{sj} \\ \text{subject to} \quad X \in \mathcal{P}_n \ , \quad Y \in \mathcal{P}_n \end{cases}$$

Theorem 5.1.1. Let $X^*(m) = (x^*_{ij}(m))$ and $Y^*(m) = (y^*_{ij}(m))$ be an optimal solution of (5.1.8). Then $Y^*(m) = (X^*(m))^t$ for $m > m_o = \sum_{i=1}^{n} \sum_{j=1}^{i-1} b_{ij}$ and $X^*(m)$ solves (5.1.6) for $m > m_o$ .

Proof. Let us first prove that $Z^*(m) \equiv Y^*(m)\, X^*(m) = I$ for $m > m_o$ . Obviously $Z^*(m) \in \mathcal{P}_n$ since $X^*(m) \in \mathcal{P}_n$ and $Y^*(m) \in \mathcal{P}_n$ . Assume that $Z^*(m) \neq I$ . Then there exists at least one pair of indices $(i,j)$ , $i > j$ for which $z^*_{ij}(m) = 1$ . It follows from this that the $i^{th}$ row



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

of $Y^*(m)$ is the transpose of the $j^{th}$ column of $X^*(m)$. This implies that $y^*_{ik}(m) = x^*_{kj}(m) = 1$ for some $k$ and $y^*_{ir}(m) = x^*_{sj}(m) = 0$ for all $r$ and $s$ such that $r \neq k$, $s \neq k$. Thus

$$\left(Y^*(m)\ B(m)\ X^*(m)\right)_{ij} = \sum_{r=1}^{n} \sum_{s=1}^{n} y^*_{ir}(m)\ b_{rs}(m)\ x^*_{sj}(m)$$

$$= b_{kk}(m) \geq m$$

Hence if $m > m_o$, then the $(i,j)^{th}$ component of the permuted matrix $Y^*(m)\ B(m)\ X^*(m)$ is strictly greater than $m_o$ and thus $\ell(X^*(m), Y^*(m)) > m_o = \ell(I,I)$ which contradicts the optimality of $X^*(m)$ and $Y^*(m)$ relative to (4.2.8). Thus we have established $Y^*(m)X^*(m) = I$ for $m > m_o$ which implies $Y^*(m) = (X^*(m))^t$ for $m > m_o$. It follows from this that

$$\min \{\ell_m(X^t,X) | X \in \mathcal{P}_n\} \leq \ell_m\left((X^*(m))^t, X^*(m)\right)$$

$$= \min \{\ell_m(Y,X) | X \in \mathcal{P}_n,\ Y \in \mathcal{P}_n\}$$

and by the obvious relation

$$\min \{\ell_m(X^t,X) | X \in \mathcal{P}_n\} \geq \min \{\ell_m(Y,X) |\ X \in \mathcal{P}_n,\ Y \in \mathcal{P}_n\}$$

we have

$$\ell_m\left((X^*(m))^t, X^*(m)\right) = \min \{\ell_m(X^t,X) | X \in \mathcal{P}_n\}$$

i.e., $X^*(m)$ solves (5.1.6). Q.E.D.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

5.2 A Location Problem on a Rectangular Network

Let $P_i$, $i=1,\ldots,m$ be the given points (cities) on a rectangular network whose location is given by $(p_i,q_i)$ relative to some coordinate system. We will consider the construction of a factory on this network which requires $n$ types of raw materials. Let $b_j$ be the demand for the $j^{th}$ raw material at the factory and let $a_{ij}$ and $c_{ij}$ be, respectively, the total supply and the unit price of $j^{th}$ raw material at city $P_i$. Let $f_j$ be the cost of transporting the unit amount of $j^{th}$ raw material per unit distance.

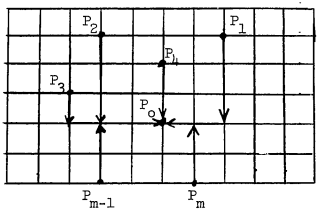



Figure 5.2.1

Our problem is to choose the location $P_o$ of the factory so as to minimize the total cost to satisfy the demands for $n$ types of raw materials.

Let $(x_o,y_o)$ be the location of $P_o$ to be determined and let $u_{ij}$ be the amount of $j^{th}$ raw material to be purchased at city $i$. Then $u_{ij}$ has to satisfy the constraints

$$(5.2.1) \qquad \begin{cases} \sum_{i=1}^{m} u_{ij} \geq b_j & j=1,\ldots,n \\ 0 \leq u_{ij} \leq a_{ij} & i=1,\ldots,m \ ; \ j=1,\ldots,n \ . \end{cases}$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The total purchasing cost $C_p$ associated with $u_{ij}$ is given by

$$C_p = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} u_{ij} \tag{5.2.2}$$

If we assume that the distance $d_i$ between $(x_o, y_o)$ and $(p_i, q_i)$ is defined by (see [W-1] for references):

$$d_i = |p_i - x_o| + |q_i - y_o| \tag{5.2.3}$$

then the total transportation cost $C_T$ is given by

$$C_T = \sum_{i=1}^{m} \sum_{j=1}^{n} f_j u_{ij} d_i = \sum_{i=1}^{m} \sum_{j=1}^{n} f_j u_{ij} \{|p_i - x_o| + |q_i - y_o|\} \tag{5.2.4}$$

So our problem is to

$$\begin{cases} \underset{x_o, y_o, u}{\text{minimize}} \; C(x_o, y_o, u) \\ \qquad = \displaystyle\sum_{i=1}^{m} \sum_{j=1}^{n} \left[ c_{ij} u_{ij} + f_j u_{ij} \left( |p_i - x_o| + |q_i - Y_o| \right) \right] \\ \text{subject to} \quad \displaystyle\sum_{i=1}^{m} u_{ij} \geq b_j \;, \quad j=1,\ldots,n \\ \qquad\qquad 0 \leq u_{ij} \leq a_{ij} \;, \quad i=1,\ldots,m \;;\; j=1,\ldots,n \end{cases} \tag{5.2.5}$$

Let us introduce the auxiliary variables $x_{i1}, x_{i2}, y_{i1}$ and $y_{i2}$, $i=1,\ldots,m$ in a standard manner, i.e., let

$$\begin{cases} x_{i1} - x_{i2} = p_i - x_o \;, \quad x_{i1} \geq 0 \;,\; x_{i2} \geq 0 \;,\; x_{i1} x_{i2} = 0 \\ \\ y_{i1} - y_{i2} = q_i - y_o \;, \quad y_{i1} \geq 0 \;,\; y_{i2} \geq 0 \;,\; y_{i1} y_{i2} = 0 \;. \end{cases} \tag{5.2.6}$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Then $|p_i - x_o| = x_{i1} + x_{i2}$ and $|q_i - y_o| = y_{i1} + y_{i2}$ . Let

$$u_j = \begin{bmatrix} u_{1j} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ u_{mj} \end{bmatrix} , \quad c_j = \begin{bmatrix} c_{1j} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ c_{mj} \end{bmatrix} , \quad a_j = \begin{bmatrix} a_{1j} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ a_{mj} \end{bmatrix} , \quad j=1,\ldots,n$$

(5.2.7)

and

$$x_\ell = \begin{pmatrix} x_{1\ell} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_{m\ell} \end{pmatrix} , \quad y_\ell = \begin{pmatrix} y_{1\ell} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ y_{m\ell} \end{pmatrix} , \quad \ell=1,2; \quad e_m = \begin{pmatrix} 1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{pmatrix} , \quad p = \begin{pmatrix} p_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ p_m \end{pmatrix} , \quad q = \begin{pmatrix} q_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ q_m \end{pmatrix} , \quad b = \begin{pmatrix} b_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ b_n \end{pmatrix}$$

(5.2.8)

Let

$$\begin{cases} U_j = \{u \in R^m | e_m^t u \geq b_j \ , \ 0 \leq u \leq a_j\} \ , \quad j=1,\ldots,n \\ X = \{x=(x_o,x_1,x_2) \in R^{2m+1} | x_o e_m + x_1 - x_2 = p \ , \quad x_1 \geq 0 \ , \ x_2 \geq 0\} \\ Y = \{y=(y_o,y_1,y_2) \in R^{2m+1} | y_o e_m + y_1 - y_2 = q \ , \quad y_1 \geq 0 \ , \ y_2 \geq 0\} \end{cases}$$

(5.2.9)

Then our problem (5.2.5) is equivalent to

(5.2.10)
$$\begin{cases} \text{minimize} \quad \mathcal{C}(x,y,u_1 \ldots u_n) \\ \qquad \sum_{j=1}^{n} \left\{ c_j + f_j(x_1 + x_2 + y_1 + y_2) \right\}^t u_j \\ \text{subject to} \\ \qquad u_j \in U_j \quad ; \quad j=1,\ldots,n \\ \qquad x \in X \ , \ y \in Y \\ \qquad x_1^t x_2 = 0 \ , \ y_1^t y_2 = 0 \ . \end{cases}$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Let us consider a bilinear programming problem associated with (5.2.10)

$$
(5.2.11)\qquad
\begin{cases}
\text{minimize} \quad \tilde{C}(x,y,u_1,\ldots,u_n) \\
\qquad\qquad = \displaystyle\sum_{j=1}^{n} \left\{ c_j + f_j(x_1 + x_2 + y_1 + y_2) \right\}^t u_j \\
\text{subject to} \quad u_j \in U_j \ , \quad j=1,\ldots,n \ ; \\
\qquad\qquad x \in X \ , \ y \in Y
\end{cases}
$$

Theorem 5.2.1. If $f_j \geq 0$ for $j=1,\ldots,n$ , then $(x_o^*, y_o^*)$ and $u_j^*$, $j=1,\ldots,n$ associated with any optimal solution of (5.2.11) gives an optimal location of the factory and the optimal quantities to be transported.

Proof. Let $(x_o^*, x_1^*, x_2^*, y_o^*, y_1^*, y_2^*, u_1^*, \ldots, u_n^*)$ be an optimal solution of (5.2.11). The theorem trivially holds if $(x_1^*)^t x_2^* = (y_1^*)^t y_2^* = 0$ . So we will assume that this is violated, i.e., $(x_1^*)^t x_2^* \neq 0$ and/or $(y_1^*)^t y_2^* \neq 0$ . Let $\bar{x} = (\bar{x}_o, \bar{x}_1, \bar{x}_2)$ be a vector defined by

$$
(5.2.12)\qquad
\begin{cases}
\bar{x}_o = x_o^* \\
\bar{x}_{i1} = \begin{cases} x_{i1}^* & \text{if} \quad x_{i1}^* x_{i2}^* = 0 \\ x_{i1}^* - x_{i2}^* & \text{if} \quad x_{i1}^* \geq x_{i2}^* > 0 \\ 0 & \text{if} \quad x_{i2}^* > x_{i1}^* > 0 \end{cases} \quad , \quad i=1,\ldots,m \\
\bar{x}_{i2} = \begin{cases} x_{i2}^* & \text{if} \quad x_{i1}^* x_{i2}^* = 0 \\ x_{i2}^* - x_{i1}^* & \text{if} \quad x_{i2}^* \geq x_{i1}^* > 0 \\ 0 & \text{if} \quad x_{i1}^* > x_{i2}^* > 0 \end{cases} \quad , \quad i=1,\ldots,m
\end{cases}
$$

and let $\bar{y} = (\bar{y}_o, \bar{y}_1, \bar{y}_2)$ be defined analogously. It is easy to see that

$$
C(\bar{x}, \bar{y}, u_1^*, \ldots, u_n^*) \leq C(x^*, y^*, u_1^*, \ldots, u_m^*)
$$

since $f_j \geq 0$ , $u_j^* \geq 0$ for $j=1,\ldots,n$ . It follows from the definition



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

that $\bar{x} \in X$ , $\bar{y} \in Y$ . Hence $(\bar{x},\bar{y},u_1^*,\ldots,u_n^*)$ is also an optimal solution of (5.2.11). Moreover, we have $\bar{x}_1^t \bar{x}_2 = 0$ , $\bar{y}_1^t \bar{y}_2 = 0$ (by definition), so that $(\bar{x},\bar{y},u_1^*,\ldots,u_n^*)$ relative to (5.2.5) follows from this and the relation

$$\begin{aligned}
&\min \{C(x,y,u_1,\ldots,u_n) \mid u_j \in U_j \text{ , } x \in X \text{ , } y \in Y \text{ , } x_1^t x_2 = 0, \; y_1^t y_2 = 0\} \\
&\quad \geq \min\{C(x,y,u_1,\ldots,u_n) \mid u_j \in U_j \text{ , } x \in X \text{ , } y \in Y\} \\
&\quad = C(\bar{x},\bar{y},u_1^*,\ldots,u_n^*) \text{ .}
\end{aligned}$$

So $(\bar{x}_o,\bar{y}_o)$ gives an optimal location of the factory. But by definition, $\bar{x}_o = x_o^*$ , $\bar{y}_o = y_o^*$ and the optimality of $(x_o^*,y_o^*)$ has been established. The optimality of $u_j^*, j=1,\ldots,n$ follows immediately.

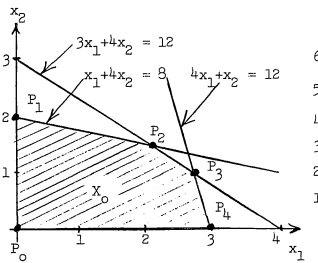
Q.E.D.

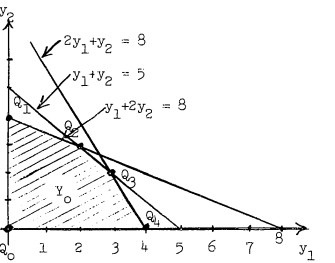


Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

## APPENDIX A. NUMERICAL EXAMPLE

$$
\left\{
\begin{array}{l}
\text{maximize} \quad (-1,1)\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + (1,0)\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} + (x_1,x_2)\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \\
\text{subject to} \\
\qquad \begin{pmatrix} 1 & 4 \\ 4 & 1 \\ 3 & 4 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leq \begin{pmatrix} 8 \\ 12 \\ 12 \end{pmatrix}, \quad \begin{pmatrix} 2 & 1 \\ 1 & 2 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \leq \begin{pmatrix} 8 \\ 8 \\ 5 \end{pmatrix} \\
\qquad (x_1,x_2) \geq 0 \quad , \quad (y_1,y_2) \geq 0
\end{array}
\right.
$$




Step 1. (Initialization). $\varphi_o = \varphi'_o = -\infty$ . $\epsilon = \delta = 0$ , $p = j = 1$.

$x^{01} = (0,0)$ , $y^{01} = (0,0)$ ; $(P_o , Q_o)$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Cycle 1

Step 2 (Suboptimization)

$$\arg\max\{\varphi(x,y^{01}) \mid x \in X_o\} = (0,2)^t \equiv x^{1,1} \qquad (P_1)$$

$$\arg\max\{\varphi(x^{1,1},y) \mid y \in Y_o\} = (0,4)^t \equiv y^{1,1} \qquad (Q_1)$$

$$\arg\max\{\varphi(x,y^{1,1}) \mid x \in X_o\} = (0,2)^t \equiv x^{2,1} = x^{1,1} \qquad (P_1)$$

$$x^{\infty,1} = (0,2)^t \ , \ y^{\infty,1} = (0,4)^t$$

Step 3 (Canonical Representation Relative to $(P_1,Q_1)$)

$$\begin{cases} \text{maximize } 10 - \left(\frac{25}{4} \ , \ \frac{5}{4}\right)\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - (2,1)\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} + (x_1,x_2)\begin{pmatrix} 15/8 & 5/8 \\ 3/8 & 1/8 \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \\ \text{subject to} \\ \begin{pmatrix} 1 & 1 \\ 2 & -1 \\ 15 & -1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \le \begin{pmatrix} 8 \\ 4 \\ 40 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 3 & -1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \le \begin{pmatrix} 8 \\ 8 \\ 2 \end{pmatrix} \\ (x_1,x_2) \ge 0 \ , \ (y_1,y_2) \ge 0 \end{cases}$$

Step 4 (Test of the Local Maximality)

$$\varphi_{1,1} = -9 \ , \ \varphi_{12} = -21/2 \ , \ \varphi_{2,1} = -8 \ , \ \varphi_{2,2} = -10$$

$\varphi_{i_o j_o} = \varphi_{2,1} = -8 < 0$; $(P_1,Q_1)$ is a vertexwise local maximum.

$$\varphi_1 = 10.$$

Step 5 (Construction of Cutting Planes)

$\varphi_o := \max\{\varphi_1,\varphi_o\} = 10$ $\quad I = \{1,2\}$ , $J = \{1,2\}$ ;

$\hat{g}^t = (25/4,5/4)$ , $\hat{h}^t = (2,1)$

$$k_{IJ} = \max\left\{\frac{15/8}{25/4 \cdot 2} \ , \ \frac{5/8}{25/4 \cdot 1} \ , \ \frac{3/8}{5/4 \cdot 2} \ , \ \frac{1/8}{5/4 \cdot 1}\right\} = \frac{3}{20}$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The cutting plane to be adjoined:

$$\frac{25}{4}x_1 + \frac{5}{4}x_2 + 2y_1 + y_2 \geq \frac{80}{3}$$

Step 6 (Optimality Test and the Computation of an Upper Bound)

$$x^{o2} = \arg\max\{\hat{g}^t x \mid x \in X_o\} = \begin{pmatrix}3\\5\end{pmatrix} \quad ; \quad (P_4)$$

$$y^{o2} = \arg\max\{\hat{h}^t y \mid y \in Y_o\} = \begin{pmatrix}4\\4\end{pmatrix} \quad ; \quad (Q_4)$$

$$\alpha_x = \hat{g}^t x^{o2} = 25 \quad , \quad \alpha_y = \hat{h}^t y^{o2} = 12 \quad , \quad \alpha = 37 > 80/3$$

$$\Phi_1 = 10 + \max\left\{0, \frac{3}{20}\left(25 - \frac{20}{3}\right)\left(12 - \frac{20}{3}\right) - \frac{20}{3}\right\} = 18$$

Cycle 2

Step 2 (Suboptimization)

$$x^{1,2} = \arg\max\{\Phi(x, y^{o2}) \mid x \in X_o\} = (3,5)^t = x^{o,2} \quad ; \quad (P_4)$$

$$y^{1,2} = \arg\max\{\Phi(x^{1,2}, y) \mid y \in Y_o\} = (4,4)^t = y^{o,2} \quad ; \quad (Q_4)$$

$$x^{\infty,2} = (3,5)^t \quad , \quad y^{\infty,2} = (4,4)^t$$

Step 3 (Canonical Representation at $(P_4, Q_4)$)

$$\text{maximize } 13 - \begin{pmatrix}\frac{15}{16} & \frac{3}{16}\end{pmatrix}\begin{pmatrix}x_1\\x_2\end{pmatrix} - \begin{pmatrix}\frac{5}{2} & 1\end{pmatrix}\begin{pmatrix}y_1\\y_2\end{pmatrix} + (x_1, x_2)\begin{pmatrix}15/64 & 5/64\\3/64 & 1/64\end{pmatrix}\begin{pmatrix}y_1\\y_2\end{pmatrix}$$

subject to

$$\begin{pmatrix}1 & 1 & & \\ 13 & -3 & & \\ 15 & -1 & & \\ & & 1 & 1 \\ & & 3 & -1 \\ & & 1 & -1 \\ \frac{25}{16} & \frac{5}{16} & \frac{5}{4} & \frac{1}{4}\end{pmatrix}\begin{pmatrix}x_1\\x_2\\y_1\\y_2\end{pmatrix} \leq \begin{pmatrix}48\\48\\80\\16\\16\\4\\31/3\end{pmatrix}$$

$$(x_1, x_2, y_1, y_2) \geq 0$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Step 4 (Test of the Local Maximality)

$$\varphi_{1,1} = -10 \quad , \quad \varphi_{1,2} = -14\frac{11}{13} \quad , \quad \varphi_{2,1} = -10 \quad , \quad \varphi_{2,2} = -13$$

$$\varphi_{i_o j_o} = \varphi_{1,1} = -10 < 0 \quad (P_4, Q_4) \text{ is a locally maximum vertex,}$$

$$\varphi_2 = 13.$$

Step 5 (Construction of Cutting Planes)

$$\varphi_o := \max\{\varphi_2, \varphi_o\} = 13 \qquad I = \{1,2\} \quad , \quad J = \{1,2\}$$

$$\hat{g}^t = (15/16,\ 3/16) \quad , \quad \hat{h}^t = (5/2,\ 1)$$

$$k_{IJ} = \max\left\{\frac{15/64}{15/16 \cdot 5/2} \quad , \quad \frac{5/64}{15/16 \cdot 1} \quad , \quad \frac{3/64}{3/16 \cdot 5/2} \quad , \quad \frac{1/64}{3/16 \cdot 1}\right\} = \frac{1}{10}$$

The cutting plane to be adjoined:

$$\frac{15}{16}x_1 + \frac{3}{16}x_2 + \frac{5}{2}y_1 + y_2 \geq 40$$

The tightening of the previously adjoined cut:

$$\tau = \max\{\sigma \mid \psi_1(\sigma) \leq \varphi_o - \varphi_1\} = \frac{88.1}{3}$$

The tighter constraint:

$$\frac{25}{16}x_1 + \frac{5}{16}x_2 + \frac{5}{4}y_1 + \frac{1}{4}y_2 \leq \frac{22.9}{3}$$

Step 6 (Optimality Test and the Computation of an Upper Bound)

$$x^{o3} = \arg\max\{\hat{g}^t x \mid x \in X_o\} = \begin{pmatrix} 8 \\ 40 \end{pmatrix}$$

$$y^{o3} = \arg\max\{\hat{h}^t y \mid y \in Y_o\} = \begin{pmatrix} 8 \\ 8 \end{pmatrix}$$

$$\alpha_x = \hat{g}^t x^{o3} = 15 \quad ; \quad \alpha_y = \hat{h}^t y^{o3} = 28$$

$$\varphi_2 = 13 + \max\left\{0 \ , \ \frac{1}{10}(15-10)(28-10) - 10\right\} = 13$$

Hence $\varphi^* \leq 13$ which implies that $(P_4, Q_4)$ gives an optimal pair of extreme points and $\varphi^* = 13$.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Computation of $\alpha$ -- An alternative way to check the optimality

$$\begin{cases} \text{maximize} \quad f(x_1,x_2,y_1,y_2) = \frac{15}{16}x_1 + \frac{3}{16}x_2 + \frac{5}{2}y_1 + y_2 \\ \text{subject to} \\ \begin{pmatrix} 1 & 1 & & \\ 13 & -3 & & \\ 15 & -1 & & \\ & & 1 & 1 \\ & & 3 & -1 \\ & & 1 & -1 \\ 25/16 & 3/16 & 5/4 & 1/4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{pmatrix} \leq \begin{pmatrix} 48 \\ 48 \\ 80 \\ 16 \\ 16 \\ 4 \\ 23/3 \end{pmatrix} \\ (x_1, x_2, y_1, y_2) \geq 0 \end{cases}$$

Obviously

$$\begin{aligned} f(x_1,x_2,y_1,y_2) &= \frac{15}{16}x_1 + \frac{3}{16}x_2 + \frac{5}{2}y_1 + y_2 \\ &\leq 4\left(\frac{25}{16}x_1 + \frac{3}{16}x_2 + \frac{5}{4}y_1 + \frac{1}{4}y_2\right) \end{aligned}$$

Hence

$$\begin{aligned} \alpha &= \max f(x_1,x_2,y_1,y_2) \\ &\leq 4 \times 22.9/3 = 30.5 < 40 = \tau \end{aligned}$$

and $\varphi_o$ gives an optimal solution.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

APPENDIX B. THE TRANSFORMATION OF A GENERAL BLP INTO A STANDARD FORM ([D-1], pp.86 ~ 88)

Assume that we are given a general BLP:

$$
(B.1)\quad
\begin{cases}
\text{maximize} \quad \varphi(x_1,x_2,y_1,y_2) = c^t \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + d^t \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} + (x_1^t, x_2^t)\, C \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \\
\text{subject to} \quad A_{11}x_1 + A_{12}x_2 \le a_1 \quad \cdots\cdots\cdots(1) \\
A_{21}x_1 + A_{22}x_2 \ge a_2 \quad \cdots\cdots\cdots(2) \\
A_{31}x_1 + A_{32}x_2 = a_3 \quad \cdots\cdots\cdots(3) \\
B_{11}y_1 + B_{12}y_2 \le b_1 \quad \cdots\cdots\cdots(4) \\
B_{21}y_1 + B_{22}y_2 \ge b_2 \quad \cdots\cdots\cdots(5) \\
B_{31}y_1 + B_{32}y_2 = b_3 \quad \cdots\cdots\cdots(6) \\
x_1 \ge 0\,,\ x_2 \text{ unrestricted},\ y_1 \ge 0\,,\ y_2 \text{ unrestricted}
\end{cases}
$$

Let us consider the transformation of (B.1) into a standard form

(i) Multiply -1 to (2) and (5);

(ii) Replace (3) and (5) by the equivalent inequalities

$$
\begin{array}{cc}
A_{31}x_1 + A_{32}x_2 \le a_3 & B_{31}y_1 + B_{32}y_2 \le b_3 \\
-e_{A_3}^t A_{31}x_1 - e_{A_3}^t A_{32} \le -e_{A_3}^t a_3 & -e_{B_3}B_{31}y_1 - e_{B_3}B_{32}y_2 \le -e_{B_3}b_3
\end{array}
$$

where $e_{A_3}$, $e_{B_3}$ are the vectors of 1's of appropriate dimensions.

(iii) Represent $x_{2j}$($j^{th}$ component of $x_2$) and $y_{2k}$($k^{th}$ component of $y_2$) as follows:

$$
\begin{aligned}
x_{2j} &= x'_{2j} - x'_{20} \qquad x'_{2j} \ge 0 \text{ for all } j \text{ and } x'_{20} \ge 0 \\
y_{2k} &= y'_{2k} - y'_{20} \qquad y'_{2k} \ge 0 \text{ for all } k \text{ and } y'_{20} \ge 0\,.
\end{aligned}
$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

By this substitution, (B.1) is reduced to a BLP in standard form

$$(B.2)\qquad \begin{cases} \text{maximize} & \varphi(x,y) = c^t x + d^t y + x^t C y \\ \text{subject to} & Ax \leq a \ , \ x \geq 0 \\ & By \leq b \ , \ y \geq 0 \end{cases}$$

where

$$A = \begin{pmatrix} A_{11} & A_{12} & -A_{12}e_2 \\ -A_{21} & -A_{22} & A_{22}e_2 \\ A_{31} & A_{32} & -A_{32}e_2 \\ -e_3^t A_{31} & -e_3^t A_{32} & e_3^t A_{32} e_2 \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2' \\ x_0' \end{pmatrix}, \quad a = \begin{pmatrix} a_1 \\ -a_2 \\ a_3 \\ -e_3^t a_3 \end{pmatrix}$$

$$B = \begin{pmatrix} B_{11} & B_{12} & -B_{12}e_2 \\ -B_{21} & -B_{22} & B_{22}e_2 \\ B_{31} & B_{32} & -B_{32}e_2 \\ -e_3^t B_{31} & -e_3^t B_{32} & e_3^t B_{32} e_2 \end{pmatrix}, \quad y = \begin{pmatrix} y_1 \\ y_2' \\ y_0' \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ -b_2 \\ b_3 \\ -e_3^t b_3 \end{pmatrix}$$



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

REFERENCES

[A-1] Altman, M., "Bilinear Programming", Bulletin d'Academie Polonaise des Sciences, Serie des Sciences Math. Astr. et Phys. 19(9), 741-746 (1968).

[B-1] Balas, E., "An Additive Algorithm for Solving Linear Programs with Zero-One Variables", J. ORSA. 13(4), 517-546 (1965).

[B-2] Balinski, M. L., "An Algorithm for Finding All Vertices of Convex Polyhedral Sets", J. Soc. Indust. Appl. Math. 9(1), 72-88 (1961).

[B-3] Benders, J. F., "Partitioning Procedures for Solving Mixed-Variable Programming Problems", Numerische Mathematik 4, 238-252 (1962).

[C-1] Cabot, A. V., and Francis, R. L., "Solving Certain Nonconvex Quadratic Minimization Problems by Ranking Extreme Points", J. ORSA. 18(1), 82-86 (1970).

[C-2] Charnes, A., "Constrained Games and Linear Programming", Proc. of the National Academy of Sciences, 639-641 (1953).

[C-3] Charnes, A., and Cooper, W. W., "Nonlinear Power of Adjacent Extreme Point Methods in Linear Programming", Econometrica 25, 132-153 (1957).

[C-4] Cottle, R. W., "The Principal Pivoting Method of Quadratic Programming", in Mathematics of the Decision Sciences, Vol. I, American Mathematical Society, Providence, R.I. (1968).



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[C-5] Cottle, R. W., and Dantzig, G. B., "Complementary Pivot Theory of Mathematical Programming", Linear Algebra and its Applications 1(1), 103-125 (1968).

[C-6] Cottle, R. W., and Ferland, J. A., "Matrix-Theoretic Criteria for the Quasi-Convexity and Pseudo-Convexity of Quadratic Functions", Technical Report No. 70-6, Dept. of Operations Research, Stanford University (1970).

[C-7] Cottle, R. W., and Ferland, J. A., "On Pseudo-Convex Function on Nonnegative Variables", Technical Report No. 70-9, Dept. of Operations Research, Stanford University (1970).

[C-8] Cottle, R. W., and Mylander, W. C., "Ritter's Cutting Plane Method for Nonconvex Quadratic Programming", in Integer and Nonlinear Programming (J. Abadie, ed.) North Holland, Amsterdam (1970).

[C-9] Candler, W., and Townsley, R. J., "The Maximization of a Quadratic Function of Variables Subject to Linear Inequalities", Management Science 10(3), 515-523 (1964).

[D-1] Dantzig, G. B., Linear Programming and Extensions, Princeton University Press, Princeton, N. J. (1963).

[D-2] Dantzig, G. B., "Solving Two-Move Games with Perfect Information", RAND Report P-1459, Santa Monica, California (1958).

[D-3] Dantzig, G. B., "Large Scale Linear Programming", in Mathematics of Decision Sciences, Vol. I, American Mathematical Society, Providence, R.I. (1968).

[D-4] Dantzig, G. B., "Reduction of a 0-1 Integer Program to a Bilinear Separable Program and to a Standard Complementary Problem", Unpublished Note (July 27, 1971).



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[E-1] Eaves, B. C., "The Linear Complementarity Problem in Mathematical Programming", Technical Report No. 69-4, Dept. of Operations Research, Stanford University (1969).

[E-2] Eaves, B. C., and Zangwill, W. I., "Generalized Cutting Plane Algorithms", Technical Report, Dept. of Operations Research, Stanford University (1969).

[F-1] Ford, L., Jr., and Fulkerson, D. R., Flows in Network, Princeton University Press, Princeton, N. J. (1962).

[G-1] Geoffrion, A. M., "Elements of Large-Scale Mathematical Programming, Part I and II", Management Science 16(11), 652-691 (1970).

[G-2] Geoffrion, A. M., "Generalized Benders' Decomposition", Working Paper No. 159, Western Management Science Institute, University of California, Los Angeles, California (1970).

[G-3] Gomory, R. E., "An Algorithm for Integer Solutions to Linear Programs", in Recent Advances in Mathematical Programming (R. L. Graves and P. Wolfe, eds.) (1963).

[H-1] Hu, T. C., "Multi-Commodity Network Flows", J. ORSA 11(3), 344-360 (1963).

[K-1] Klee, V., "A Class of Linear Programming Problems Requiring a Large Number of Iterations", in Mathematics of Decision Sciences, Vol. I, American Mathematical Society, Providence, R.I. (1968).

[K-2] Kuhn, H. W., and Tucker, A. W., "Nonlinear Programming", in Second Berkeley Symposium on Mathematical Statistics and Probability, (J. Neyman, ed.), University of California Press, Berkeley, California (1951).



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[L-1] Lasdon, L. S., Optimization Theory for Large Systems, MacMillan, New York (1970).

[L-2] Lemke, C. E., and Howson, J. T., "Equilibrium Points of Bimatrix Games", J. Soc. Indust. Appl. Math. 12(2), 413-423 (1964).

[M-1] Mangasarian, O. L., "Equilibrium Points of Bimatrix Games", J. Soc. Indust. Appl. Math. 12(4), 778-780 (1964).

[M-2] Mangasarian, O. L., and Stone, H., "Two-Person Nonzero-Sum Games and Quadratic Programming", J. Math. Anal. and Appl. 9, 348-355 (1964).

[M-3] Mills, H., "Equilibrium Points in Finite Games", J. Soc. Indust. Appl. Math. 8(2), 397-402 (1960).

[M-4] Mylander, W. C., "Nonconvex Quadratic Programming by a Modification of Lemke's Method", RAC-TP-414, Research Analysis Corporation, McLean, Virginia (1971).

[N-1] Nash, J. F., "Non Cooperative Games", Ann. Math. 54, 286-295 (1951).

[R-1] Ritter, K., "Stationary Points of Quadratic Maximum-Problem", Z. Wahrscheinlichkeitstheorie verw. Geb., 4, 149-158 (1965).

[R-2] Ritter, K., "A Method for Solving Maximum Problems with a Nonconcave Quadratic Objective Function", Z. Wahrscheinlichkeitstheorie, verw. Geb., 4, 340-351 (1966).

[R-3] Rosen, J. B., "Existence and Uniqueness of Equilibrium Points for Concave N-Person Games", Econometrica 33(3), 520-534 (1965).

[R-4] Rosen, J. B., "Gradient Projection Method for Nonlinear Programming: Part I, Linear Constraints", J. Soc. Indust. Appl. Math. 8(1), 181-217 (1960).



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[R-5] Raghavachari, M., "On Connections between Zero-one Integer Programming and Concave Programming under Linear Constraints", J. ORSA. 17(4), 680-684 (1969).

[S-1] Shanno, D. F., and Weil, R. L., "'Linear' Programming with Absolute Value Functionals", J. ORSA. 19(1), 120-124 (1971).

[S-2] Simmonard, M., Linear Programming, Prentice Hall, Englewood Cliffs, N. J. (1966).

[T-1] Thompson, G. L., Tonge, F. M., and Zionts, S., "Techniques for Removing Nonbinding Constraints and Extraneous Variables from Linear Programming", Management Science, 12(7), 588-608 (1966).

[T-2] Tomlin, J. A., "Minimum-Cost Multi-Commodity Network Flows", J. ORSA. 14(1), 45-51 (1966).

[T-3] Tui, H., "Concave Programming under Linear Constraints", Soviet Math., 1437-1440 (1964).

[W-1] Wesolowsky, G. O., and Love, R. F., "The Optimal Location of New Facilities Using Rectangular Distances", J. ORSA. 19(1), 124-130 (1971).

[W-2] Willoughby, R., (ed.) Sparse Matrix Proceedings, RA 1(11707), International Business Machine Co., Thomas J. Watson Research Center, N. Y. (1969).

[W-3] Wolfe, P., "Methods of Nonlinear Programming", in Nonlinear Programming, (J. Abadie, ed.), John Wiley and Sons, N. Y., (1967).

[W-4] Wolfe, P., "The Simplex Method for Quadratic Programming", Econometrica 27(3), 382-398 (1959).



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[Z-1] Zwart, P., "Nonlinear Programming: Counterexample to Global Optimization Algorithms Proposed by Ritter and Tui", Technical Report C00-1493-32, Dept. of Applied Mathematics and Computer Science, Washington University (1969).



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.