APPLICATIONS OF BILINEAR PROGRAMMING

Hiroshi Konno

November 1975

WP-75-154

Working·Papers are not intended for distribution outside of IIASA, and are solely for discussion and information purposes. The views expressed are those of the author, and do not necessarily reflect those of IIASA.

Applications of Bilinear Programming

Hiroshi Konno

CONTENTS

1. Introduction

This paper is addressed to some of the more important applications of bilinear programming which is a technique for solving a special class of nonconvex quadratic programming problems with the following structure:

$$\min_{x_1,x_2} \{c_1^t x_1 + c_2^t x_2 + x_1^t Q x_2 | A_1 x_1 = b_1, x_1 \geq 0, A_2 x_2 = b_2, x_2 \geq 0\} \tag{1.1}$$

where $c_i \varepsilon R^{n_i}$, $b_i \varepsilon R^{m_i}$, $A_i \varepsilon R^{m_i \times n_i}$, $i = 1,2$, $Q \varepsilon R^{n_1 \times n_2}$ are constants and $x_i \varepsilon R^{n_i}$, $i = 1,2$ are variables. We will refer to this as a bilinear program in standard minimization form. Corresponding to the above,

$$\min_{x_1,x_2} \{c_1^t x_1 + c_2^t x_2 + x_1^t Q x_2 | A_1 x_1 \geq b_1, x_1 \geq 0, A_2 x_2 \geq b_2, x_2 \geq 0\} \tag{1.2}$$

will be referred to as a bilinear program in canonical minimization form. Bilinear programs in standard and canonical maximization form will be defined anologously. As in the case of linear programming, bilinear program with general mixed equality and inequality constraints can be reduced to a standard form and to a canonical form as long as the linear constraints with respect to x_1 and x_2 are separable with each other.

Several papers have appeared since 1971 dealing with the algorithms to solve this class of problem or its equivalent, among which Konno [13], [15], Gallo-Ülkücü [7], Falk [5] are notable. Recently, the author implemented his algorithm on CYBER 74 to get encouraging numerical results [13]. At the same time, he established the finite convergence of his cutting plane algorithm [15] with the incorporation of facial cut introduced by Majthay and Whinston [19]. Now that there is a workable algorithm, we will pursue further to show the appliciability of bilinear programming to real world problems. In fact, the existence of many practical problems which are naturally put into the structure of bilinear program motivated the author's work in algorithm.

Before going into typical applications, we will briefly summarize the relationship of bilinear program (BLP) to other groups of mathematical programming problems.

First of all, BLP is a very straightforward extension of linear programs (LP) (see e.g.[3]).

$$\min_{x} \{c^t x \mid Ax = b,\ x \geq 0\} \tag{1.3}$$

where c, $x \in R^n$, $b \in R^m$, $A \in R^{m\times n}$ and c is a fixed cost vector. If we want to vary c as well as x in a polyhedral convex set, say,

$$C = \{c \in R^n \mid A'c = b',\ c \geq 0\}$$

then the problem becomes a BLP where $c_1 = c_2 = 0$ and Q is an $n \times n$ identity matrix in (1.1). We will refer to this problem

$$\min \{c^t x \mid Ax = b,\ x \geq 0,\ A'c = b',\ c \geq 0\} \tag{1.4}$$

as an extended linear program (ELP) in standard minimization form. We will discuss several examples of ELP in Chapters 2 and 3.

Secondly, there is a similar but entirely different class of problems called a generalized linear program (GLP) introduced by Dantzig and Wolfe [3]:

$$\min_{(c_j,a_j,x_j)} \left\{ \sum_{j=1}^{n} c_j x_j \,\middle|\, \sum_{j=1}^{n} a_j x_j = b,\ x_j \geq 0,\ \begin{pmatrix} c_j \\ a_j \end{pmatrix} \in C_j \subset R^{m+1}\ ,\ j = 1,\ldots,n \right\} \tag{1.5}$$

where $a_j \in R^m$ and C_j is a closed convex set, $j = 1,\ldots,n$. This program is the source of the famous decomposition algorithm. Here the column vectors $\begin{pmatrix} c_j \\ a_j \end{pmatrix}$ as well as x_j's are variables and each column $\begin{pmatrix} c_j \\ a_j \end{pmatrix}$ is allowed to move in a closed convex set C_j independendly of each other. This independence property distinguishes itself from BLP and it is quite essential for GLP algorithm (decomposition algorithm) to work (see [3]). We will look into the relationship between GLP and BLP in section 5.1 and show that the special case of a non-standard generalized linear program, i.e., a GLP some of whose variables x_j are not restricted in sign, is essentially a BLP.

Thirdly, it will be shown in section 4.1, that the so-called linear max-min problem (LMMP):

$$\min_{x \in X} \max_{y \in Y} \{p_1^t x + p_2^t y | B_1 x + B_2 y \geq b\} \tag{1.6}$$

where X and Y are polyhedral convex sets, can be converted into a BLP by taking the partial dual of (1.6) with respect to Y under some regularity condition. This problem was treated by Falk [5] as well as by Dantzig [4] and Konno [17]. It will be shown in section 3.1 that LMMP has several applications with game theoretic flavour.

Fourthly, it is possible, at least theoretically to transform the problem with complementarity condition

$$\min \{c_1^t x_1 + c_2^t x_2 | A_1 x_1 = b_1, x_1 \geq 0, A_2 x_2 = b_2, x_2 \geq 0, x_1^t x_2 = 0\} \tag{1.7}$$

into a BLP by putting $x_1^t x_2$ term into the objective function as follows:

$$\min \{c_1^t x_1 + c_2^t x_2 + M x_1^t x_2 | A_1 x_1 = b_1, x_1 \geq 0, A_2 x_2 = b_2, x_2 \geq 0\}$$

where M is a large positive constant. (1.7) was analyzed by Ibaraki [10] and by Konno [17]. We will briefly touch on this topic in section 5.2.

Finally, it has been proved in [14] that the minimization of concave quadratic function subject to linear constraints (CQP):

$$\min \{2c^t x + x^t Q x | Ax = b, x \geq 0\} \tag{1.8}$$

where Q is a symmetric negative semi-definite matrix, can be converted into a BLP:

$$\min \{c^t u + c^t v + u^t Q v | Au = b, u \geq 0, Av = b, v \geq 0\} \tag{1.9}$$

The relationship between (1.8) and (1.9) has been fully discussed elsewhere [14], where we will show that (i) if x* is optimal for (1.8), then (u,v) = (x*,x*) is optimal for (1.9) and (ii) if (u*,v*) is optimal for (1.9), then both u* and v* are optimal for (1.8). Also it has been shown how to exploit the symmetric structure of (1.9) to improve the cutting plane algorithm developed in [13]. It is well known that CQP is closely related to 0-1 integer program and there˜ore BLP is indirectly related to 0-1 integer program.

The following figure briefly summarizes the relationship among various problems cited above, the details of which will be discussed in full scope

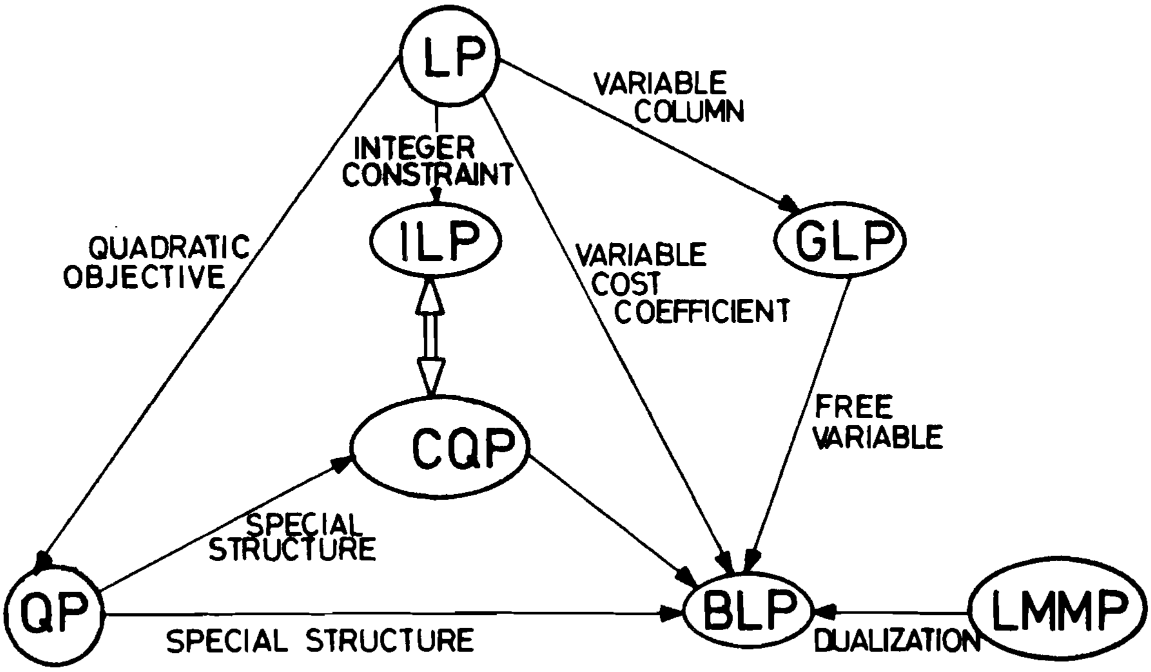


FIGURE 1.1

at the appropriate places.

In the following chapters, we will choose some of the typical examples from the various areas of applications and discuss them in some detail. We tried to pick up, among others, the problems which are of practical, theoretical and computational interests. This paper is an elaboration of the author's earlier paper [17] with significant revisions and additions.

2. Extended Linear Programs

In this chapter we will discuss two typical examples of BLP with very natural physical and economical interpretation. The first one is minimization of the transportation cost which is the cross product of quantity and distance. The second one is the application in decision analysis in which we want to minimize the cross product of probability measure and weights of importance.

2.1 Location-Allocation Problems

There is a large amount of literature under the title of location-allocation theory. (See, for example, reference [24]). Suppose we are given

a) a set of m points distributed in the plane
b) a vector value to be attached to each point
c) a set of indivisible centroids without predetermined locations

then the location-allocation problem in its most general form is to find locations for m centroids and an allocation of vector value associated with n points to some centroid so as to minimize the total cost. Here, we will present one original example of this type of problems which is put into the structure of BLP in a very natural way.

(a) Single Factory Case

Let there be m cities P_i, $i = 1,\ldots,m$ on a plane. P_i is located at (p_i,q_i) relative to some coordinate system. We are going to construct a factory somewhere on this plane. This factory needs b_j units of n different materials M_j, $j = 1,\ldots,n$.

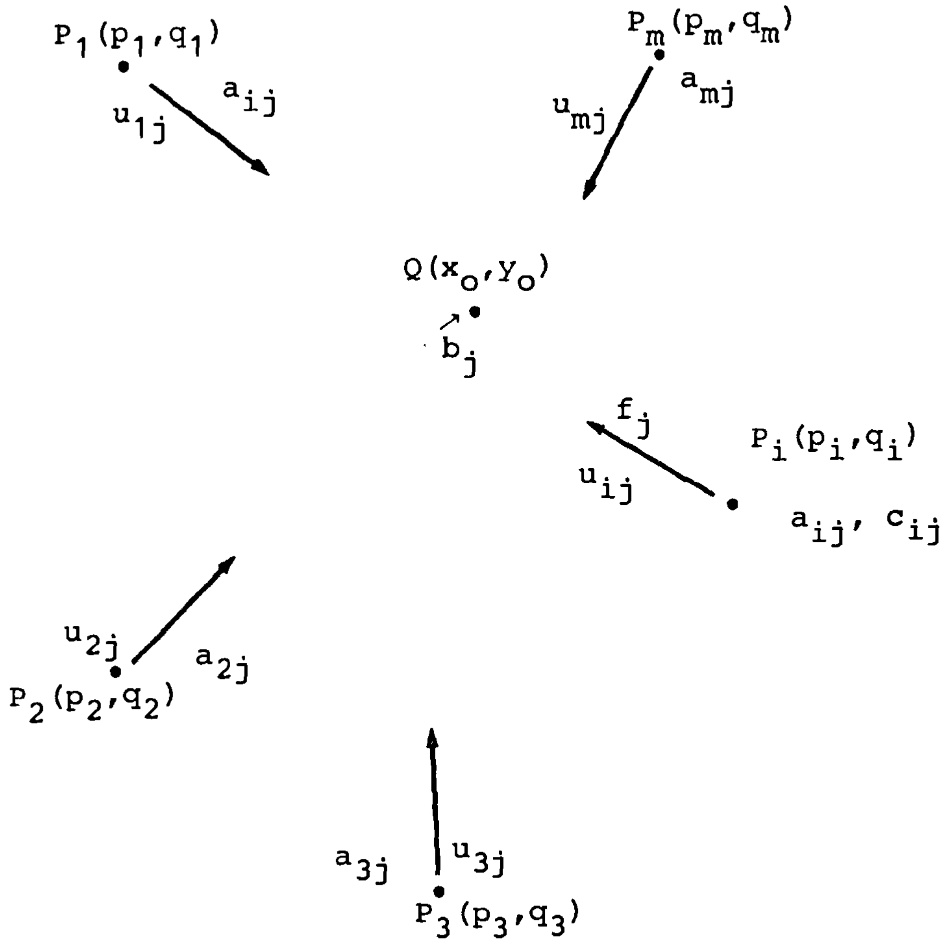


Figure 2.1.1 Single Factory Case

b_j = requirement for M_j

u_{ij} = shipment of M_j from P_i

a_{ij} = supply of M_j at P_i

c_{ij} = unit price of M_j at P_i

f_j = unit shipment cost of M_j

Let us assume that P_i can supply a_{ij} units of M_j at the unit price c_{ij} and the unit transportation cost (per unit amount per unit distance) of M_j is given by f_j. Our concern now is to minimize the total expense which is represented by the sum of total purchasing cost and the total transportation cost.

Let $Q(x_o,y_o)$ be the location of the factory to be constructed and let u_{ij} be the amount of M_j to be purchased at P_i. Then u_{ij} has to satisfy:

$$\left\{\begin{array}{lll} \sum_{i=1}^{m} u_{ij} \geq b_j \quad , & j = 1,\ldots,n \quad , & \\ 0 \leq u_{ij} \leq a_{ij} & i = 1,\ldots,m \quad ; & j = 1,\ldots,n \quad . \end{array}\right. \tag{2.1.1}$$

Total purchasing cost C_p is obviously given by:

$$C_p = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}u_{ij} \tag{2.1.2}$$

and total transportation cost C_T is given by

$$C_T = \sum_{i=1}^{m} \sum_{j=1}^{n} f_j \cdot u_{ij}\, d(P_i,Q) \tag{2.1.3}$$

where $d(P_i,Q)$ is the distance between P_i and Q.

i) <u>Manhattan Distance</u>

If the distance $d(P_i,Q)$ is given by 1 norm i.e.,

$$d(P_i,Q) = d_1(P_i,Q) \equiv |p_i - x_o| + |q_i - y_o| \tag{2.1.4}$$

then the total cost C is given by

$$C = \sum_{i=1}^{m} \sum_{j=1}^{n} \left[c_{ij}u_{ij} + f_ju_{ij}(|p_i-x_o| + |q_i-y_o|)\right] \tag{2.1.5}$$

By introducing auxiliary variables, x_{i1} and y_{i1} satisfying

$$\begin{array}{ll} x_{i1} - x_{i2} = p_i - x_o & x_{i1} \geq 0,\ x_{i2} \geq 0,\ x_{i1}x_{i2} = 0,\ i = 1,\ldots,m \quad , \\ y_{i1} - y_{i2} = q_i - y_o & y_{i1} \geq 0,\ y_{i2} \geq 0,\ y_{i1}y_{i2} = 0,\ i = 1,\ldots,m \quad . \end{array} \tag{2.1.6}$$

the absolute value terms can be written as:

$$|p_i - x_o| = x_{i1} + x_{i2}$$

$$|q_i - y_o| = y_{i1} + y_{i2} \qquad (2.1.7)$$

So the problem now is to

$$\text{minimize } C = \sum_{i=1}^{m} \sum_{j=1}^{n} u_{ij} \left[c_{ij} + f_j(x_{i1} + x_{i2} + y_{i1} + y_{i2})\right]$$

$$\text{s.t.} \quad \sum_{i=1}^{m} u_{ij} \geq b_j \quad , \quad j = 1,\ldots,n$$

$$0 \leq u_{ij} \leq a_{ij} \quad i = 1,\ldots,m, \quad j = 1,\ldots,n \quad ,$$

$$x_{i1} - x_{i2} + x_o = p_i$$

$$1 = 1,\ldots,m$$

$$y_{i1} - y_{i2} + y_o = q_i$$

$$x_{il} \geq 0 \quad , \quad y_{il} \geq 0 \quad i = 1,\ldots,m, \quad 1 = 1,2,$$

$$x_{il} \cdot x_{i2} = 0 \quad , \quad y_{i1} \cdot y_{i2} = 0 \quad , \; i = 1,\ldots m \; . \qquad (2.1.8)$$

It is straightforward to show that the optimal solution of the associated bilinear program without the orthogonality condition in (2.18) automatically satisfy the orthognality property if $f_j \geq 0$, $j = 1,\ldots,n$ and hence the problem can be solved by applying the algorithm developed in [13].

ii) Euclidean Distance

If on the other hand, the distance $d(P_i,Q)$ is given by 2 norm, i.e.,

$$d(P_i,Q) = d_2(P_i,Q) \equiv \sqrt{(p_i - x_o)^2 + (q_i - y_o)^2} \qquad (2.1.9)$$

then the problem becomes:

$$\left\{\begin{aligned} &\text{minimize } C = \sum_{i=1}^{m} \sum_{j=1}^{n} \left[c_{ij} + f_j \sqrt{(p_i - x_o)^2 + (q_i\ y_o)^2}\right] u_{ij} \\ &\sum_{i=1}^{m} u_{ij} \geq b_j \qquad j = 1,\ldots,n \ , \\ &0 \leq u_{ij} \leq a_{ij} \quad i = 1,\ldots,m \ , \quad j = 1,\ldots,n \ . \end{aligned}\right. \tag{2.1.10}$$

to which we can apply a modified version of the BLP algorithm.

(b) Multi-Factory Case

Let us consider here the multi factory version of the problem treated in the previous section. The basic setting of the problem is the same as before except

(i) $K(\geq 1)$ factories F_k, $k = 1,\ldots,K$ have to be constructed

(ii) each factory is produving L different types of commodities C_ℓ, $\ell = 1,\ldots,L$

(iii) each product has to be shipped to the demand points i.e., to m cities.

Let

u_{ij}^{k} : the amount of M_j to be purchased at P_i and shipped to F_k

$x_{i\ell}^{k}$: amount of C_ℓ to be shipped to P_i from F_k

b_j^k : amount of M_j required at F_k

a_{ij} : maximum supply of M_j at P_i

c_{ij} : unit price of M_j at P_i

d_ℓ^k : amount of C_ℓ produced at F_k

$e_{i\ell}$: demand for C_ℓ at P_i

(p_i, q_i) : location of P_i

(x_k, y_k) : location of F_k

$d(P_i, F_k)$: distance between P_i and F_k

f_j : unit transportation cost of M_j

g_ℓ : unit transportation cost of C_ℓ

The total cost is now given by

$$C = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{K} c_{ij} u_{ij}^{k} + \sum_{i=1}^{m} \sum_{k=1}^{K} (\sum_{\ell=1}^{L} g_l x_{i\ell}^{k} + \sum_{j=1}^{n} f_j u_{ij}^{k}) d(P_i, F_k) \tag{2.1.11}$$

Also u_{ij}^k and $x_{i\ell}^k$ have to satisfy:

$$\sum_{i=1}^{m} u_{ij}^{k} \geq b_j^k \quad , \quad j = 1,\ldots,n \quad , \quad k = 1,\ldots,K$$

$$\sum_{k=1}^{K} u_{ij}^{k} \leq a_{ij} \quad , \quad i = 1,\ldots,m \quad , \quad j = 1,\ldots,n$$

$$\sum_{i=1}^{m} x_{i\ell}^{k} \leq d_\ell^k \quad , \quad \ell = 1,\ldots,L \quad , \quad k = 1,\ldots,K$$

$$\sum_{k=1}^{K} x_{i\ell}^{k} \geq e_{i\ell} \quad , \quad i = 1,\ldots,m \quad , \quad k = 1,\ldots,K$$

$$u_{ij}^{k} \geq 0 \quad , \quad x_{i\ell}^{k} \geq 0 \quad . \tag{2.1.12}$$

Hence now the problem is to minimize (2.1.11) subject to (2.1.12) which is a BLP if d(•,•) is defined by 1 norm.

We assumed here that there are no material flows between the factories to be constructed. Should there be such a flow, then the problem can no longer be formulated in the framework of bilinear programming.

2.2 Applications to Decision Analysis

Suppose a decision maker is facing a problem of choosing the 'best' among m possible alternatives A_i, $i = 1,\ldots,m$ in the stochastic environment where n possible events E_j, $j = 1,\ldots,n$ takes place with probability p_{ij} when A_i is chosen.

Let us suppose also that there are K independent attributes (objectives) T_k, $k = 1,\ldots, K$, each of which has weight (degree of importance) w_k. Also let us assume that the utility associated with the triple (A_i, E_j, T_k), is given by a_{ij}^k and that the overall utility of the decision maker is additive, i.e., the expected utility u_i obtained by choosing A_i is given by

$$u_i = \sum_{k=1}^{k} \sum_{j=1}^{n} w_k p_{ij} a_{ij}^k \quad . \qquad (2.2.1)$$

Given the constants w_k, p_{ij}, a_{ij}^k, we can choose the optimal alternative by simply comparing u_i, $i = 1,\dots,m$.

It sometimes happens, however, due to the lack of information that the quantities w_k, $k = 1,\dots,K$ and p_{ij}, $i = 1,\dots,m$; $j = 1,\dots,n$ are not known precisely. Typically, the analyst has to interview the decision maker to estimate the weight of relative importance w_k of T_k and it sometimes happens that we only have interval estimates

$$\underline{w}_k \leq w_k \leq \bar{w}_k \quad , \qquad k = 1,\dots,K \quad .$$

where $\underline{w}_k$ and $\bar{w}_k$ are given constants (see [23]).

Similar situation applies as well to the probability measure p_{ij}^k. Let us suppose here that

$$\underline{p}_{ij} \leq p_{ij} \leq \bar{p}_{ij} \quad , \qquad i = 1,\dots,m \quad ; \quad j = 1,\dots,n \quad .$$

$$\sum_{j=1}^{n} p_{ij} = 1 \quad , \qquad i = 1,\dots,m \quad .$$

where $\underline{p}_{ij}$ and $\bar{p}_{ij}$ are given constants.

In this case, the optimal alternative will not be uniquely determined. However, some of the alternatives may be eliminated as inefficient ones as follows:

Let

$$W = \{(w_1,\dots,w_k) | \underline{w}_k \leq w_k \leq \bar{w}_k\} \quad . \qquad (2.2.2)$$

$$P_i = \{(p_{i1},\dots,p_{in}) | \underline{p}_{ij} \leq p_{ij} \leq \bar{p}_{ij} \quad , \quad j = 1,\dots,n \quad ; \quad \sum_{j=1}^{n} p_{ij} = 1\} \quad . \qquad i = 1,\dots,m \qquad (2.2.3)$$

which we assume to be nonempty.

Let

$$\underline{u}_i = \min \{ \sum_{k=1}^{K} \sum_{j=1}^{n} w_k p_{ij} a_{ij}^k \mid w \varepsilon W,\ p_i \varepsilon P_i \} \qquad (2.2.4)$$

$$\bar{u}_i = \max \{ \sum_{k=1}^{K} \sum_{j=1}^{n} w_k p_{ij} a_{ij}^k \mid w \varepsilon W,\ p_i \varepsilon P_i \} \qquad (2.2.5)$$

It is obvious that if $\underline{u}_r > \bar{u}_s$, then A_r is preferred to A_s and A_s can be eliminated from the candidates of optimal alternatives. Similarly, if

$$u_r,s \equiv \min \{ \sum_{k=1}^{K} \sum_{j=1}^{n} w_k (p_{rj} a_{rj}^k - p_{sj} a_{sj}^k) \mid w \varepsilon W, p_r \varepsilon P_r, p_s \varepsilon P_s \} > 0 \qquad (2.2.6)$$

then A_r is preferred to A_s and A_s can be eliminated. Problems (2.2.4)(2.2.5) and (2.2.6) are all bilinear programs with a very special structure. Let us take for example (2.2.4) suppressing index i:

$$\min \sum_{k=1}^{k} \sum_{j=1}^{n} a_j^k p_j w_k$$

$$\text{s.t.} \quad \sum_{j=1}^{n} p_j = 1 \ , \quad \underline{p}_j \leq p_j \leq \bar{p}_j \ , \quad j = 1,\ldots,n \ ;$$

$$\underline{w}_k \leq w_k \leq \bar{w}_k \ , \quad k = 1,\ldots,K \ . \qquad (2.2.7)$$

The following theorem characterizes the form of an optimal solution of (2.2.7) which is guaranteed to exist since W and P are non-empty compact convex sets.

Theorem 2.2.1

Let $\hat{w}_k$, $k = 1,\ldots,K$; $\hat{p}_j$, $j = 1,\ldots,n$ be an optimal solution of (2.2.7). Then $\hat{w}_k$ is equal to $\underline{w}_k$ or $\bar{w}_k$ for all k. Also, p_j is equal to $\underline{p}_j$ or $\bar{p}_j$ except possibly for one index j_0.

Proof

W and P are bounded polyhedral convex sets. Hence by the fundamental theorem BLP [13], there exists an optimal solution $(\hat{w},\hat{p})$ where $\hat{w}$ and $\hat{p}$ are extreme points of W and P, respectively.

It is easy to see that any extreme point of W and P has the property stated in the theorem. $||$

Using this theorem, we can construct a simple enumeration technique by fixing w_k equal to $\underline{w}_k$ or $\bar{w}_k$. Also it may be more appropriate in some cases to normalize w_k, $k = 1,\ldots, K$ to satisfy the condition $\sum_{k=1}^{K} w_k = 1$, as well as p_j in which case we still have a bilinear program with somewhat more complicated structure. We will not, however, go into details about these any further. For the background material of decision analysis the readers are referred to Keeney-Raiffa [12] and to Sarin [23].

3. Bilinear Assignment Problems

3.1 Introduction

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 j. Then the problem of maximizing total profit can be formulated as follows:

$$\begin{aligned} &\text{maximize} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij}x_{ij} \\ &\text{s.t.} \quad \sum_{j=1}^{n} x_{ij} = 1 \quad , \qquad i = 1,\ldots,n \quad ; \\ &\qquad \sum_{i=1}^{n} x_{ij} = 1 \quad , \qquad j = 1,\ldots,n; \qquad\qquad (3.1.1) \\ &\qquad x_{ij} = 0 \text{ or } 1 \quad , \qquad i,j = 1,\ldots,n. \end{aligned}$$

This problem is called a standard linear assignment problem. Let

$$Z_n = \{(z_{ij})\varepsilon R^{n^2} \mid \sum_{j=1}^{n} z_{ij} = 1,\ i = 1,\ldots n;\ \sum_{i=1}^{n} z_{ij} = 1,$$
$$j = 1,\ldots,n;\ z_{ij} \geq 0,\ i,j = 1,\ldots,n\} \qquad (3.1.2)$$

and

$$Z_n^I = \{(z_{ij})\ R^{n^2} \mid (z_{ij})\varepsilon Z_n,\ z_{ij} = 0 \text{ or } 1,\ i,j = 1,\ldots,n\} \qquad (3.1.3)$$

It it well known that all the extreme points of Z_n belong to Z_n^I and hence 0-1 constraints on the variables in (3.1.1) can be replaced by nonnegativity constraints, so that (3.1.1) is equivalent to the following linear program (see e.g. [3]).

$$\text{maximize } \{ \sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij}x_{ij} \mid (x_{ij})\varepsilon Z_n\} \ . \qquad (3.1.4)$$

As a natural extension to the above, the following quadratic assignment problem:

$$\text{maximize } \{ \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{r=1}^{n} \sum_{s=1}^{n} a_{ij}^{rs} x_{ij} x_{rs} \mid (x_{ij}) \varepsilon Z_n^I \} \qquad (3.1.5)$$

has been considered by several authors. Unfortunately, however, Z_n^I cannot be replaced by Z_n in this case and the problem is difficult to solve in general (see [8],[22]).

Let us introduce here a new class of assignment problems:

$$\text{maximize } \{ \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{r=1}^{n} \sum_{s=1}^{n} a_{ij}^{rs} x_{ij} y_{rs} \mid (x_{ij}) \varepsilon Z_n^I,\ (y_{rs}) \varepsilon Z_n^I \} \qquad (3.1.6)$$

We will call (3.1.6) a standard bilinear assignment problem. As in the linear case, this problem has a very nice structure as shown in the following theorem.

Theorem 3.1.1

(3.1.6) is equivalent to

$$\text{maximize } \{ \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{r=1}^{n} \sum_{s=1}^{n} a_{ij}^{rs} x_{ij} y_{rs} \mid (x_{ij}) \varepsilon Z_n,\ (y_{rs}) \varepsilon Z_n \} \qquad (3.1.7)$$

Proof

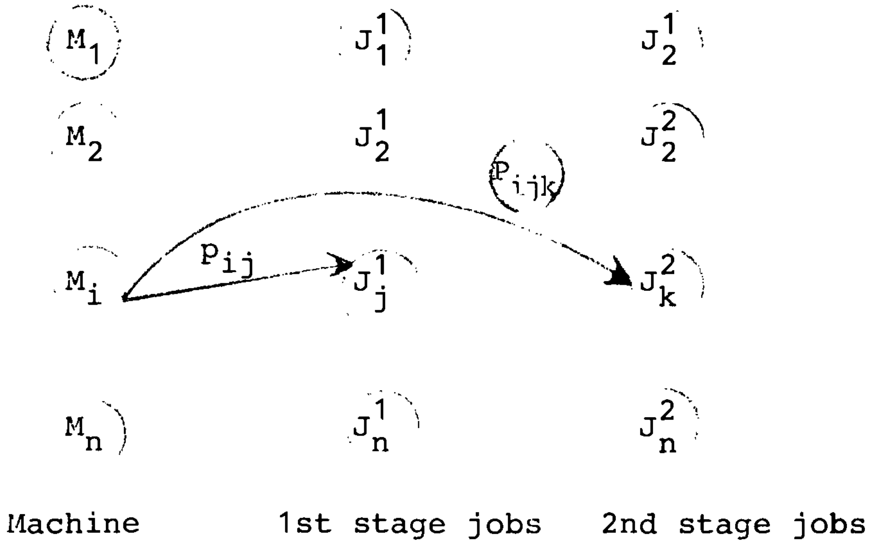
(3.1.7) is a bilinear program in standard form and Z_n is non-empty and bounded. Hence by the fundamental theorem of BLP (see [13]) this problem has an optimal solution $(\hat{x}_{ij})$ and $(\hat{y}_{rs})$, both of which are extreme points of Z_n. Also, any extreme point of Z_n belongs to Z_n^I and hence the pair $(\hat{x}_{ij})$, $(\hat{y}_{rs})$ is an optimal solution of (3.1.6). ||

The implication of this theorem is that we can obtain a locally optimal solution by the repeated use of simplex algorithm, i.e., by the augmented mountain climbing algorithm described in [13], which is not the case for general quadratic assignment problems. We can obtain, if we like, a global optimal solution by applying the full version of the cutting plane algorithm proposed in [15]. We will discuss three typical examples of bilinear assignment problems in the following sections.

3.2 Multi-Stage Markovian Assignment Problem and Three Dimensional Assignment Problem

3.2.1 Multi-Stage Markovian Assignment Problem

Let us consider the two stage version of the standard linear assignment problem. Let there be n machines M_i, $i = 1,\ldots,n$ as before and each one of these machines has to be assigned to one and only one of the n jobs J_j^1, $j = 1,\ldots,n$ at the first stage and one and only one of the n jobs J_k^2, $k = 1,\ldots,n$ at the second stage. Here $\{J_j^1\}_1^n$ and $\{J_k^2\}_1^n$ need not be the same set of jobs. We will assume that the outcome associated with $M_i \rightarrow J_j^1$ at first stage is given by p_{ij}. Due to necessary machine setup to different kind of jobs, the outcome associated with $M_i \rightarrow J_k^2$ at the second stage depends upon which job M_i was assigned to in the first stage. Let p_{ijk} be the outcome associated with $M_i \rightarrow J_k^2$ given $M_i \rightarrow J_j^1$.



For $\ell = 1,2$, let us define variables x_{ij}^ℓ as follows:

$$x_{ij}^\ell = \begin{cases} 1 \text{ if } M_i \text{ is assigned to } J_j^\ell \text{ at stage } \ell \\ 0 \quad \text{otherwise} \end{cases} \tag{3.2.1}$$

Then the two stage optimization problem is formulated as

$$\begin{cases} \max \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} (p_{ij} + p_{ijk}) x_{ij}^{1} x_{ik}^{2} \\ \text{s.t.} \quad (x_{ij}^{1}) \varepsilon Z_n^I \quad , \quad (x_{ij}^{2}) \varepsilon Z_n^I \quad . \end{cases} \qquad (3.2.2)$$

where Z_n^I is defined by (3.1.3), which is equivalent to

$$\begin{cases} \max \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} (p_{ij} + p_{ijk}) x_{ij}^{1} x_{ik}^{2} \\ \text{s.t.} \quad (x_{ij}^{1}) \varepsilon Z_n \quad , \quad (x_{ik}^{2}) \varepsilon Z_n \end{cases} \qquad (3.2.3)$$

as we have shown in Theorem 3.1.1.

This approach applies to the general L stage assignment problem as long as the inter-stage dependence is Markovian, i.e., if the outcome of a particular assignment at stage t is dependent only upon the current assignment and the assignment of the previous stage.

Let q_{ijk}^{ℓ} be the outcome of the assignment $M_i \rightarrow J_k^{\ell}$ at stage ℓ given $M_i \rightarrow J_j^{\ell-1}$ at stage ℓ-1. Then the problem becomes

$$\begin{cases} \text{maximize} \sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij} x_{ij}^{1} + \sum_{\ell=2}^{L} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} q_{ijk}^{\ell} \; x_{ij}^{\ell-1} x_{ik}^{\ell} \\ \quad \text{s.t.} \quad (x_{ij}^{\ell}) \varepsilon Z_n^I \quad , \quad \ell = 1,\ldots,L \end{cases} \qquad (3.2.4)$$

Let $x^{\ell} = (x_{ij}^{\ell})$, $\ell = 1,\ldots,L$ and let

$$Q_i^{\ell} = \begin{bmatrix} q_{i11}^{\ell} & q_{i12}^{\ell} & q_{i1n}^{\ell} \\ \cdot & & \\ \cdot & & \\ \cdot & & \\ \cdot & & \\ q_{in1}^{\ell} & & q_{inn}^{\ell} \end{bmatrix} \swarrow \begin{matrix} i = 1,\ldots,n; \\ \ell = 1,\ldots,L; \end{matrix} \qquad (3.2.5)$$

$$\ell = 1,\ldots,L \searrow \qquad Q^{\ell} = \begin{bmatrix} Q_1^{\ell} & & 0 \\ & Q_2^{\ell} & \\ & \ddots & \\ 0 & & Q_n^{\ell} \end{bmatrix} \qquad (3.2.6)$$

For simplicity, let L be an even integer and let

$$u = \begin{pmatrix} x^1 \\ x^3 \\ \cdot \\ \cdot \\ \cdot \\ x^{L-1} \end{pmatrix} \qquad v = \begin{pmatrix} x^2 \\ x^4 \\ \cdot \\ \cdot \\ \cdot \\ x^L \end{pmatrix} \tag{3.2.7}$$

Then we can rewrite (3.2.4) in a vector form

$$\begin{cases} \text{maximize } p^t x^1 + u^t Q v \\ \text{s.t.} \quad u \varepsilon \prod_1^{L/2} Z_n^I \ , \quad v \varepsilon \prod_1^{L/2} Z_n^I \end{cases} \tag{3.2.8}$$

where

$$Q = \begin{bmatrix} Q_2 & & & \\ Q_3^t & Q_4 & & \\ & \ddots & \ddots & \\ & & Q_{L-1}^t & Q_L \end{bmatrix} \tag{3.2.9}$$

It is again easy to show that (3.2.8) is equivalent to the bilinear program in standard form:

$$\begin{cases} \text{maximize } p^t x^1 + u^t Q v \\ \text{s.t.} \quad u \varepsilon \prod_1^{L/2} Z_n \ , \quad v \varepsilon \prod_1^{L/2} Z_n \ . \end{cases} \tag{3.2.10}$$

3.2.2 Three Dimensional Assignment Problem

Let there be m research assistants R_i, i = 1,...,m, n scientists S_j, j = 1,...,n and p projects P_k, k = 1,...,p. Let a_{ijk} be the productivity of the combination (R_i, S_j, P_k). We will assume that each research assistant has to be assigned to one and only one combination of S_j and P_k. Also each scientist S_j can be assigned at most b_j times and each project has to have at least c_k combinations of scientists and research assistants. This is a typical example of a three dimensional assignment problem (see [22]).

Introducing three indexed variables x_{ijk} with the implication

$$x_{ijk} = \begin{cases} 1 \text{ if the assignment } (R_i,S_j,P_k) \text{ takes place} \\ 0 \text{ otherwise} \end{cases} \qquad (3.2.11)$$

the problem can be formulated as follows:

$$\begin{cases} \text{maximize} & \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{p} a_{ijk} \, x_{ijk} & \\ \text{s.t.} & \sum_{j=1}^{n} \sum_{k=1}^{p} x_{ijk} = 1 \;, & i = 1,\ldots,m \;; \\ & \sum_{i=1}^{m} \sum_{k=1}^{p} x_{ijk} \leq b_j \;, & j = 1,\ldots,n \;; \\ & \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ijk} \geq c_k \;, & k = 1,\ldots,p \;; \\ & x_{ijk} = 0 \text{ or } 1 \;, \; \forall i,j,k \;. & \end{cases} \qquad (3.2.12)$$

Contrary to the two index case, the constraint matrix (3.2.12) is not totally unimodular and hence cannot be solved by the simplex method. This problem has mnp variables and m+n+p constraints.

It has been shown, however, by A.M. Frieze [6] that this problem is reformulated as a bilinear program. Let

$$y_{ij} = \begin{cases} 1 \text{ if } R_i \text{ is assigned to } S_j \\ 0 \text{ otherwise} \end{cases} \qquad (3.2.13)$$

$$z_{ik} = \begin{cases} 1 \text{ if } R_i \text{ is assigned to } P_k \\ 0 \text{ otherwise} \end{cases} \qquad (3.2.14)$$

Then since R_i is assigned only once, we have the relationship

$$x_{ijk} = y_{ij} \cdot z_{ik} \quad . \qquad \forall_{i,j,k} \qquad (3.2.15)$$

as long as the following conditions are satisfied:

$$\sum_{j=1}^{n} y_{ij} = 1 \quad , \qquad i = 1,\ldots,m \quad ,$$

$$\sum_{j=1}^{n} z_{ij} = 1 \quad , \qquad i = 1,\ldots,m \quad . \qquad (3.2.16)$$

Hence (3.2.12) is equivalent to

$$\max \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{p} a_{ijk}\, y_{ij}\, z_{ik} \quad .$$

$$\text{s.t.} \quad \sum_{j=1}^{n} y_{ij} = 1 \quad , \qquad i = 1,\ldots,m \quad ;$$

$$\sum_{i=1}^{m} y_{ij} \leq b_j \quad , \qquad j = 1,\ldots,n \quad ;$$

$$\sum_{k=1}^{p} z_{ik} = 1 \quad , \qquad i = 1,\ldots,m \quad ;$$

$$\sum_{i=1}^{m} z_{ik} \geq c_k \quad , \qquad k = 1,\ldots,p \quad .$$

$$y_{ij}, z_{ik} = 0 \text{ or } 1 \quad . \qquad \forall_{i,j,k} \quad . \qquad (3.2.17)$$

where we used the relationship

$$\sum_{i=1}^{m} \sum_{k=1}^{p} x_{ijk} = \sum_{i=1}^{m} \left(y_{ij} \sum_{k=1}^{p} z_{ik} \right) = \sum_{i=1}^{m} y_{ij} \quad .$$

etc. Let

$$Y = \{ (y_{ij}) \mid \sum_{j=1}^{p} y_{ij} = 1, \; i = 1,\ldots,m; \; \sum_{i=1}^{m} y_{ij} \leq b_j, \; j = 1,\ldots,n\}$$

$$Z = \{ (z_{ik}) \mid \sum_{k=1}^{p} z_{ik} = 1, \; i = 1,\ldots,m; \; \sum_{i=1}^{m} z_{ik} \geq c_k, \; k = 1,\ldots,p\} \qquad (3.2.18)$$

The constraint matrices defining Y and Z are totally unimodular i.e., the determinants of all the basis matrices are +1 or -1 and hence (3.2.17) is equivalent to the following BLP.

$$\max \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{p} a_{ijk} \, y_{ij} \, z_{ik}$$

$$\text{s.t.} \quad (y_{ij}) \in Y \quad , \qquad (z_{ik}) \in Z \quad . \qquad (3.2.19)$$

This problem has 2m+n+p constraints (m more than (3.2.12)) and only m(n+p) variables compared with mnp of (3.2.12). Also it is important to notice that we can apply mountain climbing algorithm to obtain a local solution in the framework of standard linear programming procedure, which is not the case for (3.2.12).

3.3 Reduction of a Sparse Matrix into an Almost Triangular Matrix

3.3.1 Some Examples

Let A = (a_{ij}) be a given n × n matrix which contains many zero entries. It is sometimes desirable to permute the rows and/or columns of this matrix and rearrange it into an almost triangular matrix. Let us give here some of the typical examples.

a) Solution of a System of Equations Ax = b

Let A be a nonsingular n × n matrix. It is well known that the system of equations $Ax = b$ can be solved by simple substitution when A is either lower or upper triangular matrix. Even if A is not triangular, the efficiency of Gaussian elimination will be greatly enhanced if there are fewer entries $a_{ij} \neq 0$, $i > j$ (see [26]). So it is very desirable to find a way to permute rows and columns to obtain an (almost) triangular matrix.

b) Structuring of an Input-Output System

Let there be n industrial sectors S_i each producing commodity C_i, $i = 1,\ldots,n$. Let us assume that a_{ij} is the amount of C_i required when S_j produces a unit amount of C_j. Then $n \times n$ matrix $A = (a_{ij})$ represents the relationship among S_i, $i = 1,\ldots,n$. Let us say that S_j is independent of S_i if $a_{ij} = 0$. Suppose we want to put an ordering on S_i, $i = 1,\ldots,n$ based upon this notion of independence.

It is obvious that if we can arrange A into an upper-triangular matrix by simultaneous permutation of rows and columns, then it gives

$$\begin{array}{cc} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} & \\ \left[\begin{array}{ccccc} x & x & & & x \\ & x & & & x \\ x & x & x & x & x \\ x & x & & x & x \\ & & & & x \end{array}\right] & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \end{array} \Longrightarrow \begin{array}{cc} \begin{array}{ccccc} 3 & 4 & 1 & 2 & 5 \end{array} & \\ \left[\begin{array}{ccccc} x & x & x & x & x \\ & x & x & x & x \\ & & x & x & x \\ & & & x & x \\ & & & & x \end{array}\right] & \begin{array}{c} 3 \\ 4 \\ 1 \\ 2 \\ 5 \end{array} \end{array}$$

a total ordering among the industries. This is not always possible. However, if we can find an ordering which produces an almost triangular matrix then it would serve as a reasonable ordering.

c) Ranking n Players in a Contest

Suppose we are given the result of a contest played by n persons P_i, $i = 1,\ldots,n$. Each game is played between two players and each pair of players played this game at *most* once. Let

$a_{ij} = 1$ if, and only if P_i won over P_j and 0 otherwise. Then the ranking of these n players become the same problem as b) above and the approach stated above would give one reasonable ranking.

3.3.2 Mathematical Formulation

Let $B = (b_{ij})$ be a binary $n \times n$ matrix associated with $A = (a_{ij})$:

$$b_{ij} = \begin{cases} 1 \text{ if } a_{ij} \neq 0 \\ 0 \text{ if } a_{ij} = 0 \end{cases} \qquad (3.3.1)$$

and let

$$P_n = \{X = (x_{ij}) \varepsilon R^{n\times n} | \sum_{j=1}^{n} x_{ij} = 1, \quad i = 1,\ldots,n; \\ \sum_{j=1}^{n} x_{ij} = 1, \; j = 1,\ldots,n\}$$

$$P_n^I = \{X = (x_{ij}) \varepsilon R^{n\times n} | X \varepsilon P_n; \; x_{ij} = 0 \text{ or } 1, \; i, \; j = 1,\ldots,n\}$$

P_n and Z_n are essentially the same set except that P_n is considered as a set in $R^{n\times n}$ rather than in R^{n^2}. The same remark applies to P_n^I and Z_n^I as well.

a) Arbitrary Permutation of Rows and Columns

Any combination of row and column permutations of B can be represented as XBY where $X\varepsilon P_n^I$ and $Y\varepsilon P_n^I$. The number of nonzero entries in the lower triangular portion of the matrix after the permutation is given by $\sum_{j=1}^{n} \sum_{i=j+1}^{n} (XBY)_{ij}$. Therefore to minimize this quantity, we have to solve

$$\text{minimize } \{ \sum_{j=1}^{n} \sum_{i=j+1}^{n} (XBY)_{ij} | X \varepsilon P_n^I, \; Y \varepsilon P_n^I \} \qquad (3.3.2)$$

or

$$\begin{cases} \text{minimize} \sum_{j=1}^{n} \sum_{i=j+1}^{n} \sum_{r=1}^{n} \sum_{s=1}^{n} x_{ir} b_{rs} y_{sj} \\ \text{s.t.} \quad (x_{ij}) \in Z_n^I \ , \quad (y_{ij}) \in Z_n^I \ . \end{cases} \qquad (3.3.3)$$

which is equivalent to a standard BLP without integrality conditions

$$\begin{cases} \text{minimize} \sum_{j=1}^{n} \sum_{i=j+1}^{n} \sum_{r=1}^{n} \sum_{s=1}^{n} x_{ir} b_{rs} y_{rj} \\ \text{s.t.} \quad (x_{ij}) \in Z_n \ , \quad (y_{ij}) \in Z_n \end{cases} \qquad (3.3.3')$$

Problem a) of the previous section falls into this category.

b) Synchronized Permutations of Rows and Columns

Suppose now that the permutation of rows i and j has to be associated with the permutation of columns i and j. Then we have to solve the following problem

$$\begin{cases} \min \ \ell(X^t,X) = \sum_{j=1}^{n} \sum_{i=j+1}^{n} (X^t BX)_{ij} \\ \text{s.t.} \qquad X \in P_n^I \end{cases} \qquad (3.3.4)$$

or

$$\begin{cases} \min \ \ell(X^t,X) = \sum_{j=1}^{n} \sum_{i=j+1}^{n} \sum_{r=1}^{n} \sum_{s=1}^{n} x_{ri} b_{rs} x_{sj} \\ \text{s.t.} \qquad (x_{ij}) \in Z_n^I \end{cases} \qquad (3.3.4')$$

This is a general quadratic assignment problem and we cannot replace Z_n^I by Z_n. But it is at least theoretically possible to reduce this to a standard bilinear program as we shall see in the following.

Let

$$b_{ij}(m) = b_{ij} + m\delta_{ij} \quad , \qquad i,j = 1,\ldots,n \quad . \qquad (3.3.5)$$

where m is a positive constant and δ_{ij} is a standard Kronecker's delta, i.e.,

$$\delta_{ij} = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ otherwise} \end{cases} \qquad (3.3.6)$$

Let us consider the following bilinear program:

$$\begin{cases} \min \ell_m(Y,X) = \sum_{j=1}^{n} \sum_{i=j+1}^{n} \sum_{r=1}^{n} \sum_{s=1}^{n} y_{ir} b_{rs}(m) x_{sj} \\ \text{s.t.} \qquad X = (x_{ij}) \in P_n \quad , \qquad Y = (y_{ij}) \in P_n \quad . \end{cases} \qquad (3.3.7)$$

Theorem 3.3.1

Let $Y^*(m) = (y^*_{ij}(m))$ and $X^*(m) = (x^*_{ij}(m))$ be an optimal solution of (3.3.7). Then $X^*(m) = Y^*(m)$ for $m > m_o \equiv \sum_{j=1}^{n} \sum_{i=j+1}^{n} b_{ij}$ and $X^*(m)$ solves (3.3.4) for $m > m_o$.

Proof

First note that $X^*(m) \varepsilon P_n^I$ and $Y^*(m) \varepsilon P_n^I$. Next, let us prove

$$z^*_{ij}(m) \equiv \sum_{r=1}^{n} y^*_{ir}(m) x^*_{rj}(m) = \delta_{ij} \quad \forall\ i,\ j \quad . \qquad (3.3.8)$$

Since $X^*(m) \varepsilon P_n^I$ and $Y^*(m) \varepsilon P_n^I$, $Z^*(m) \equiv (z^*_{ij}(m)) \varepsilon P_n^I$. If (3.3.8) does not hold, then there exists at least one pair of indices (i,j), i > j for which $z^*_{ij}(m) = 1$. It follows that $y^*_{i\ell}(m) = x^*_{\ell j}(m)$ for $\ell = 1,\ldots,n$. This implies that $y^*_{ik}(m) = x^*_{kj}(m)$ 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

$$\sum_{j=1}^{n} \sum_{i=j+1}^{n} \sum_{r=1}^{n} \sum_{s=1}^{n} y^*_{ir}(m) b_{rs}(m) x^*_{sj}(m) \geqq b_{kk}(m) \geqq m_o$$

$$= \sum_{j=1}^{n} \sum_{i=j+1}^{n} \sum_{r=1}^{n} \sum_{s=1}^{n} \delta_{ir} b_{rs}(m) \delta_{sj} \quad .$$

which is a contradiction to the assumption of optimality of $Y^*(m)$ and $X^*(m)$ in (3.3.7). Thus we have established (3.3.8) and therefore $y^*_{ij}(m) = x^*_{ji}(m)$ for all i,j. It follows from this that

$$\min \{\ell_m(X^t,X) \mid X \in P_n\} \leq \ell_m(X^*(m)^t, X^*(m))$$

$$= \min \{\ell_m(Y,X) \mid X \in P_n, Y \in P_n\}$$

and by obvious relation

$$\min \{\ell_m(X^t,X) \mid X \in P_n\} \geq \min \{\ell_m(Y,X) \mid X \in P_n, Y \in P_n\}$$

we have

$$\ell_m(X^*(m)^t, X^*(m)) = \min \{\ell_m(X^t,X) \mid X \in P_n\} \quad .$$

i.e., $X^*(m)$ solves (3.3.7). ||

We have established the equivalence of (3.3.4) and (3.3.7). From the computational point of view, however, solving (3.3.7) instead of (3.3.4) would not be more attractive since (3.3.7) necessarily has many local minima because of large diagonal entries.

4. Game-Theoretic Applications

4.1 Two-Stage Game: A Linear Max-Min Problem

The concept of a two stage game was first introduced by Dantzig [4] and recently treated by the author [17] and by Falk [5]. Also the author applied this concept to analyze the current conflicting situation between resource producing countries and industrialized countries in a somewhat different context [16].

Let there be two players (nations) P_1, P_2 and let $S = \{s_1, s_2, \ldots, s_m\}$ and $T = \{t_1, t_2, \ldots, t_n\}$ be the sets of activities available to P_1 and P_2, respectively. At the first stage of the game, P_1 chooses an activity level vector $x = (x_1, \ldots, x_m) \geq 0$ which satisfies the given constraints. Let us assume for simplicity that the feasible set of activities X is defined by the system of linear inequalities:

$$X = \{x \varepsilon R^m | A_1 x \leq a_1, \; x \geq 0\} \quad . \tag{4.1.1}$$

where $A \varepsilon R^{k_1 \times m}$, $a_1 \varepsilon R^{k_1}$ are given matrix and vector.

In the second stage of the game, given the full information on the P_1's choice of $x \varepsilon X$. P_2 chooses his action y from the feaible set Y(x) which depends explicitly on x. Let us assume here that Y(x) is also defined by the system of linear inequalities:

$$Y(x) = Y \cap B_y(x) \quad . \tag{4.1.2}$$

where

$$\left.\begin{aligned} Y &= \{y \varepsilon R^n | A_2 y \leq a_2, \; y \geq 0\} \quad . \\ B_y(x) &= \{y \varepsilon R^n | B_2 y \leq b - B_1 x, \; y \geq 0\} \end{aligned}\right. \tag{4.1.3}$$

where A_2, B_2 $\varepsilon R^{k_2 \times n}$, $B_1 \varepsilon R^{k_2 \times m}$; b, $a_2 \varepsilon R^{k_2}$.

We will assume here that Y(x) is nonempty and compact for all

$x\varepsilon X$. As to the payoff structure of this game, we will assume that P_1 has to pay

$$f(x,y) = p_1^t x + p_2^t y \tag{4.1.4}$$

to P_2 when P_1 and P_2 choose $x\varepsilon X$ and $y\varepsilon Y(x)$, respectively, where $p_1 \varepsilon R^m$, $p_2 \varepsilon R^n$ are given vectors.

Given $x\varepsilon X$, P_2 naturally wants to maximize $f(x,y)$ over $y\varepsilon Y(x)$. Hence his problem is to

$$\text{maximize } \{p_2^t y \mid y\varepsilon Y(x)\} \quad . \tag{4.1.5}$$

Let $y^*(x)$ be an optimal solution for this problem, which always exists since we assumed that $Y(x)$ is nonempty and compact for all $x\varepsilon X$. The problem for P_1 is now to

$$\text{minimize } \{p_1^t x + p_2^t y^*(x) \mid x\varepsilon X\} \quad . \tag{4.1.6}$$

or equivalently,

$$\min_{x\varepsilon X} \max_{y\varepsilon Y} \{p_1^t x + p_2^t y \mid B_1 x + B_2 y \leq b\} \quad . \tag{4.1.7}$$

Falk [5] named (4.1.7) a 'linear max-min problem'.For fixed x, the maximizing part with respect to y is a linear program

$$\max \{p_2^t y \mid B_2 y \leq b - B_1 x,\ A_2 y \leq a_2,\ y \geq 0\} \quad .$$

Taking the dual of this linear program:

$$\min \{(b-B_1 x)^t z_1 + a_2^t z_2 \mid B_2^t z_1 + A_2^t z_2 \geq p_2,\ z_1 \geq 0,\ z_2 \geq 0\}$$

we get a bilinear program which is equivalent to (4.1.7)

$$\begin{cases} \min\ p_1^t x + (b-B_1 x)^t z_1 + a_2^t z_2 \quad . \\ \text{s.t. } A_1 x \leq a_1 \quad , \quad x \geq 0 \quad , \\ \quad B_2^t z_1 + A_2^t z_2 \geq p_2 \quad , \quad z_1 \geq 0 \quad , \quad z_2 \geq 0 \quad . \end{cases} \tag{4.1.8}$$

Let us consider now the linear min-max problem associated with

(4.1.7).

$$\max_{y\varepsilon Y}\min_{x\varepsilon X}\{p_1^t x + p_2^t y | B_1 x + B_2 y \leq b\} \quad . \tag{4.1.9}$$

This corresponds to the two stage game in which P_2 plays first and P_1 plays next.

It is well known that in the standard zero sum two person game where the sets of feasible strategies X and Y are given by

$$X = \{(x_1,\dots,x_m) | \sum_{i=1}^{m} x_i = 1 \quad , \quad x_i \geq 0 \quad , \quad i = 1,\dots,m\}$$

$$Y = \{(y_1,\dots,y_n) | \sum y_j = 1 \quad , \quad y_j \geq 0 \quad , \quad j = 1,\dots,n\}$$

and where the payoff is given by $x^t Ay$ where A is a given m × n matrix the famous min-max theorem [25] holds:

$$\max_{y\varepsilon Y}\min_{x\varepsilon X} x^t Ay = \min_{x\varepsilon X}\max_{y\varepsilon Y} x^t Ay \quad .$$

namely there exists an equilibrium. However, in our setting it is quite exceptional to have an equilibrium.

Let

$$B_y(x) = \{y\varepsilon R^n | B_2 y \leq b - B_1 x\}$$

$$B_x(y) = \{x\varepsilon R^m | B_1 x \leq b - B_2 y\} \quad . \tag{4.1.10}$$

The next theorem is an extended version of the one proved by Falk [5].

Theorem 4.1.1

Under the assumption we made,

$$\max_{y\varepsilon Y}\min_{x\varepsilon X}\{p_1^t x + p_2^t y | B_1 x + B_2 y \leq b\}$$

$$\geq \min_{x\varepsilon X}\max_{y\varepsilon Y}\{p_1^t x + p_2^t y | B_1 x + B_2 y \leq b\} \tag{4.1.11}$$

Proof

$$\max_{y\varepsilon Y}\min_{x\varepsilon X}\{p_1^tx + p_2^ty \mid B_1x + B_2y \leq b\}$$

$$= \max_{y\varepsilon Y}\{p_2^ty + \min_{x\varepsilon X}[p_1^tx \mid B_1x \leq b - B_2y]\}$$

$$\geq \max_{y\varepsilon Y}\{p_2^ty + \min_{x\varepsilon X} p_1^tx\}$$

$$= \min_{x\varepsilon X}\{p_1^tx + \max_{y\varepsilon Y} p_2^ty\}$$

$$\geq \min_{x\varepsilon X}\{p_1^tx + \max_{y\varepsilon Y}[p_2^ty \mid B_2y \leq b - B_1x]\}$$

$$= \min_{x\varepsilon X}\max_{y\varepsilon Y}\{p_1^tx + p_2^ty \mid B_1x + B_2y \leq b\} \quad .$$

This theorem tells us an intuitively obvious fact that [5] P_1 will lose less if he plays first in the game. In fact Falk gives an example in which strict inequality holds in (4.1.11).
Again (4.1.9) can be put into the structure of bilinear program:

$$\left\{\begin{array}{ll} & \max\ p_2^ty + (b - B_2y)^tz_1 + a_1^tz_2 \\ \text{s.t.} & A_2y \leq a_2 \ , \quad y \geq 0 \ . \\ & B_1^tz_1 + A_1^tz_2 \geq P_1 \ , \quad z_1 \geq 0 \ , \quad z_2 \geq 0 \ . \end{array}\right. \tag{4.1.12}$$

In particular, if the structure of the game is symmetric, i.e., if $p_1 = p_2 \equiv p$, $A_1 = A_2 \equiv A$, $B_1 = B_2 \equiv B$, $a_1 = a_2 \equiv a$ then (4.1.8) and (4.1.12) become:

$$\min\ \{f(u,z_1,z_2) \mid u\varepsilon U,\ (z_1,z_2)\varepsilon\ Z\} \tag{4.1.13}$$

$$\max\ \{f(u,z_1,z_2) \mid u\varepsilon U,\ (z_1,z_2)\varepsilon\ Z\} \tag{4.1.14}$$

where

$$f(u,z_1,z_2) = p^tu + (b - Bu)^tz_1 + a^tz_2 \quad .$$

$$U = \{u \mid Au \leq a,\ u \geq 0\}$$

$$Z = \{(z_1,z_2) \mid B^tz_1 + A^tz_2 \geq p,\ z_1 \geq 0,\ z_2 \geq 0\} \quad .$$

It is seen from these that only in a very exceptional situation we have the equality in (4.1.11).

4.2 An Equilibrium Solution of a Constrained Bimatrix Game

The standard bimatrix game (or non zero-sum two person game) is defined as follows [25]. Let there be two players P_1, P_2 and let $S = \{s_1 s_2, \ldots, s_m\}$ and $T = \{t_1, t_2, \ldots, t_n\}$ be the sets of actions 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. Given two matrices $A = (a_{ij})$, $B = (b_{ij})$, each player chooses the mixed strategies (or probability measure) on S and T, i.e., P_1 chooses $x \varepsilon X_o$ and P_2 chooses $y \varepsilon Y_o$ where

$$X_o = \{x \varepsilon R^m | e_m^t x = 1, \ x \geq 0\} \tag{4.2.1}$$

$$Y_o = \{y \varepsilon R^n | e_n^t y = 1, \ y \geq 0\} \tag{4.2.2}$$

where e_m and e_n are m and n dimensional vectors all of whose components are ones. Then the expected payoffs to P_1 and P_2 are given by $x^t Ay$ and $x^t By$, respectively. (Note that the two players P_1 and P_2 choose x and y simultaneously). Let us denote this game as $\Gamma(A, X_o; \ B, Y_o)$.

Definition 4.2.1

A pair of mixed strategies $(\tilde{x}, \tilde{y}) \varepsilon X_o \times Y_o$ is a Nash equilibrium point of a bimatrix game $\Gamma(A, X_o: \ B, Y_o)$ if

$$\tilde{x}^t A\tilde{y} = \max \{x^t A\tilde{y} | x \varepsilon X_o\} \tag{4.2.3}$$

$$\tilde{x}^t B\tilde{y} = \max \{\tilde{x}^t By | y \varepsilon Y_o\} \quad . \tag{4.2.4}$$

The Nash equilibrium point implies that as long as P_1 sticks to $\tilde{x}$, then $\tilde{y}$ is the optimal mixed strategies for P_2 and vice versa. It has been shown in [2] that the problem of obtaining a Nash equilibrium point of $(A, X_o: \ B, Y_o)$ can be reduced to a linear complementarity problem (LCP):

Find: $z \varepsilon R^{m+n}$ such that $z \geq 0$, $s + Mz \geq 0$ and $z^t(s + Mz) = 0$ where

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

Also Lemke and Howson [ ] showed that their ingeneous complementary pivot algorithm generates a solution to this problem in finitely many steps.

Now let us introduce a constrained bimatrix game $\Gamma(A,X;B,Y)$ in which P_1 and P_2 have to choose their mixed strategies x an y from more general constraint sets X an Y, respectively. We will call X and Y admissible sets and $x \varepsilon X$, $y \varepsilon Y$ admissible strategies.

Definition 4.2.2

A pair of admissible strategies $(\tilde{x},\tilde{y}) \varepsilon X \times Y$ is called a Nash equilibrium point of $\Gamma(A,X: B,Y)$ if

$$\tilde{x}^t A\tilde{y} = \max \{x^t A\tilde{y} | x \varepsilon X\} \qquad (4.2.5)$$

$$\tilde{x}^t B\tilde{y} = \max \{\tilde{x}^t Ay | y \varepsilon Y\} \qquad (4.2.6)$$

The following fundamental theorem has been established by Nash [2].

Theorem 4.2.1

If both X and Y are non-empty, compact and convex, then $\Gamma(A,X; B,Y)$ has a Nash equilibrium point for arbitrary A and B.

Though this existence theorem has been known for years, no algorithm has been proposed except for the standard bimatrix game $\Gamma(A,X_0; B,Y_0)$ and for zero sum game i.e., $A + B = 0$. Let us now consider the 'constrained' bimatrix game $(A,X_1:B,Y_1)$ where

$$X_1 = \{x \varepsilon R^{m_1} | P_1 x = P_1, \ x \geq 0\} \qquad (4.2.7)$$

$$Y_1 = \{y \varepsilon R^{m_2} | P_2 y = p_2, \ y \geq 0\} \qquad (4.2.8)$$

where $P_i \varepsilon R^{\ell_i \times m_i}$, $p_i \varepsilon R^{\ell_i}$, $i = 1,2$.

We will pursue along the line developed in [20] and reduce the problem of obtaining an equilibrium point of $\Gamma(A,X_1: B,Y_1)$ into a bilinear program.

First let us state the following lemma.

Lemma 4.2.2

$(\tilde{x},\tilde{y})\varepsilon X \times Y$ is an equilibrium point of $\Gamma(A,X_1; B,Y_1)$ if and only if they satisfy the following system dual variables together with $\hat{u}\varepsilon R^{\ell_1}$ and $\hat{v}\varepsilon R^{\ell_2}$.

$$\begin{array}{llll} \tilde{x}^t A\tilde{y} - p_1^t\tilde{u} = 0 & & \tilde{x}^t B\tilde{y} - p_2^t\tilde{v} = 0 & \\ A\tilde{y} - P_1^t\tilde{u} \leq 0 & & B^t\tilde{x} - P_2^t\tilde{v} \leq 0 & \\ P_1\tilde{x} = p_1 & & P_2^t\tilde{y} = p_2 & \\ \tilde{x} \geq 0 & (4.2.8) & \tilde{y} \geq 0 & (4.2.9) \end{array}$$

Proof

The right hand sides of definitions (4.2.5) (4.2.6) when $A = X_1$ and $Y = Y_1$ are linear programs. (4.2.8) follows from (4.2.5) by the duality theorem of linear programming. (4.2.9) follows from (4.2.6) analogously. ||

Let us now introduce an associated bilinear program in view of (4.2.8) and (4.2.9)

$$\begin{array}{lll} \text{maximize } \phi(x,y,u,v) & & \\ \qquad = x^t(A+B)y - p_1^t u - p_2^t v & & \\ \text{s.t.} \quad B^t x - P_2^t v & & \leq 0 \\ \qquad P_1 x & & = p_1 \\ & Ay - P_1^t u & \leq 0 \\ & P_2 y & = p_2 \\ \qquad x \geq 0 \ , & y \geq 0 \ . & \end{array} \qquad (4.2.10)$$

Theorem 4.2.3

A necessary and sufficient condition for $(\tilde{x},\tilde{y})$ to be an equilibrium point of $\Gamma(A,X_1: B,Y_1)$ is that $(\tilde{x},\tilde{y})$ is an optimal solution of (4.2.10). Also $\phi(x,y,u,v) = 0$ at the optimum.

Proof

Let $(\tilde{x},\tilde{y})$ be an equilibrium point of $\Gamma(A,X_1; B,Y_1)$. Then by (4.2.8) and (4.2.9), $(\tilde{x},\tilde{y})$ together with $(\tilde{u},\tilde{v})$ satisfy the constraints of (4.2.10), i.e., $(\tilde{x},\tilde{y},\tilde{u},\tilde{v})$ is a feasible solution of (4.2.10). Also $\phi(\tilde{x},\tilde{y},\tilde{u},\tilde{v}) = 0$. However, by premultiplying x^t, y^t to the first and the third inequalities of (4.2.10) and using the facts that $P_1x = p_1$ and $P_2y = p_2$, we obtain the inequality.

$$0 \geq x^tAy - x^tP^tu = x^tAy - p_1^tu$$

$$0 \geq x^tBy - v^tQy = x^tBy - p_2^tv$$

and hence $\phi(x,y,u,v) \leq 0$ for all feasible solution of (4.2.10), whence $(\tilde{x},\tilde{y},\tilde{u},\tilde{v})$ is an optimal solution of (4.2.10). Let (x^*,y^*,u^*,v^*) be another optimal solution for (4.2.10). Then $\phi(x^*,y^*,u^*,v^*) = 0$. But this holds if and only if

$$(x^*)^tAy^* - p_1^tu^* = (x^*)^tBy^* - p_2^tv^* = 0 \quad .$$

so that (x^*,y^*,u^*,v^*) satisfies both (4.2.8) and (4.2.9), i.e., (x^*,y^*,u^*,v^*) is also an equilibrium point of $\Gamma(A,X_1: B,Y_1)$. ||
This theorem states that the optimal objective value of (4.2.10) is equal to zero regardless of the data of the problem. This property is quite worthwhile since we can generate a 'deep' cut at a poor local maximum point where the objective functional value is far from optimal (see [13]).

5. Miscellaneous Bilinear Programs

5.1 Non-Standard Generalized Linear Program

Generalized linear program (GLP) introduced by Dantzig and Wolfe [3] has the following problem structure:

$$\begin{aligned} \min \ & \sum_{j=1}^{n} c_j x_j \\ \text{s.t.} \ & \sum_{j=1}^{n} a_j x_j = b \ , \\ & x_j \geq 0 \ , \quad \begin{pmatrix} c_j \\ a_j \end{pmatrix} \varepsilon \, C_j \ . \end{aligned} \tag{5.1.1}$$

where $a_j \varepsilon R^m$, $c_j \varepsilon R^1$ and $C_j \subset R^{m+1}$ is a compact convex set $j = 1, \dots, n$ and maximization is with respect to $\begin{pmatrix} c_j \\ a_j \end{pmatrix}$ as well as x_j. The GLP algorithm by Dantzig and Wolfe proceeds roughly as follows:

Let $\begin{pmatrix} \tilde{c}_j^{\ell} \\ \tilde{a}_j^{\ell} \end{pmatrix} \varepsilon \, C_j$, $\ell = 1, \dots, \ell_j$, $j = 1, \dots, n$ be given.

Then we will solve the linear program:

$$\begin{aligned} \min \ & \sum_{j=1}^{n} \sum_{\ell=1}^{\ell_j} \tilde{c}_j^{\ell} x_j^{\ell} \\ & \sum_{j=1}^{n} \sum_{\ell=1}^{\ell_j} \tilde{a}_j^{\ell} x_j^{\ell} = b \\ & x_j^{\ell} \geq 0 \ , \quad \ell = 1, \dots, \ell_j \ ; \quad j = 1, \dots, n \end{aligned} \tag{5.1.2}$$

and let $\tilde{\pi} \varepsilon R^m$ be an optimal multiplier vector for this linear program.

If

$$c_j - \tilde{\pi} \, a_j \geq 0 \quad \forall \begin{pmatrix} c_j \\ a_j \end{pmatrix} \varepsilon \, C_j \ ; \quad j = 1, \dots, n \ .$$

then the current solution is optimal. If, on the other hand there is an index j and a vector $\begin{pmatrix} c_j \\ a_j \end{pmatrix} \varepsilon C_j$ for which

$c_j - \tilde{\pi} a_j > 0$, then the objective function will be improved by introducting this vector into the basis. To find out the vectors $\begin{pmatrix} c_j \\ a_j \end{pmatrix}$ for which $c_j - \tilde{\pi} a_j > 0$, we solve the following n subprograms.

$$\min \{c_j - \tilde{\pi} a_j | \begin{pmatrix} c_j \\ a_j \end{pmatrix} \varepsilon C_j\} \quad , \qquad j = 1,\ldots,n \quad . \tag{5.1.3}$$

Let $\begin{pmatrix} c_j^* \\ a_j^* \end{pmatrix}$ be its optimal solution. If $c_j^* - \tilde{\pi} a_j^* < 0$, then we will introduce it into (5.1.2) and proceed. If C_j are all polyhedral convex sets, then this algorithm will converge to the optimal solution of (5.1.1) in finitely many steps if we avoid cycling caused by degeneracy appropriately.

Now let us consider the non-standard GLP with some free variables, i.e.,

$$\begin{aligned} &\min \sum_{j=1}^{n} c_j x_j \\ &\text{s.t.} \quad \sum_{j=1}^{n} a_j x_j = b \\ &\qquad x_j \geq 0 \quad , \qquad j = 1,\ldots,\ell \quad ; \\ &\qquad x_j \gtreqless 0 \quad , \qquad j = \ell+1,\ldots,n \quad ; \\ &\qquad \begin{pmatrix} c_j \\ a_j \end{pmatrix} \varepsilon C_j \quad , \quad j = 1,\ldots,n \quad . \end{aligned} \tag{5.1.4}$$

The usual technique of replacing a free variable by two non-negative variables destroys the structure of the problem, i.e., let

$$x_j = x_{j1} - x_{j2} \quad , \quad x_{j1} \geq 0 \quad , \quad x_{j2} \geq 0 \quad , \quad j = \ell+1,\ldots,n \quad .$$

then the problem is

$$\min \sum_{j=1}^{\ell} c_j x_j + \sum_{j=\ell+1}^{n} c_{j1} x_{j1} - \sum_{j=\ell+1}^{n} c_{j2} x_{j2}$$

$$\sum_{j=1}^{\ell} a_j x_j + \sum_{j=\ell+1}^{n} a_j x_{j1} - \sum_{j=\ell+1}^{n} a_j x_{j2} = b$$

$$x_j \geq 0 \quad , \qquad j = 1,\ldots,\ell$$

$$x_{j1}, x_{j2} \geq 0 \quad , \qquad j = \ell+1,\ldots,n$$

$$\begin{pmatrix} c_j \\ a_j \end{pmatrix} \varepsilon C_j, \; j = 1,\ldots,\ell . \begin{pmatrix} c_{j1} \\ a_{j2} \end{pmatrix} = \begin{pmatrix} c_{j2} \\ a_{j2} \end{pmatrix} \varepsilon C_j, \; j = \ell+1,\ldots,n \quad .$$

(5.1.5)

Hence the columns of this problem are no longer independent and GLP algorithm in its original form would not work.

Now let us consider the simplest case of the above in which a_j's are constant and only c_j's are allowed to move in compact convex sets, i.e., closed interval in this case

$$\min \sum_{j=1}^{\ell} c_j x_j + \sum_{j=\ell+1}^{n} c_j x_j$$

$$\text{s.t.} \quad \sum_{j=1}^{\ell} a_j x_j + \sum_{j=\ell+1}^{n} a_j x_j = b$$

$$x_j \geq 0 \quad , \qquad j = 1,\ldots,\ell \quad ;$$

$$\underline{c}_j \leq c_j \leq \bar{c}_j \quad , \qquad j = 1,\ldots,n \quad . \qquad (5.1.6)$$

Since $x_j \geq 0$, $j = 1,\ldots,\ell$, it is obvious that optimal c_j's are $\bar{c}_j$'s for $j = 1,\ldots,\ell$. Hence the problem simplifies somewhat to

$$\begin{aligned} \min \quad & \sum_{j=1}^{\ell} \bar{c}_j x_j + \sum_{j=\ell+1}^{n} y_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{\ell} a_j x_j + \sum_{j=\ell+1}^{n} a_j x_j = b \\ & x_j \geq 0 \quad , \quad j = 1,\ldots,\ell \quad ; \\ & \underline{c}_j \leq y_j \leq \bar{c}_j \quad , \quad j = \ell+1,\ldots,n \quad . \end{aligned} \qquad (5.1.7)$$

We will use the standard elimination technique to obtain an expression of x_k, $k = \ell+1,\ldots,n$ with respect to x_j, $j = 1,\ldots,\ell$. Let

$$x_j = d_{jo} + \sum_{k=1}^{\ell} d_{jk} x_k \quad , \quad j = \ell+1,\ldots,n \quad . \qquad (5.1.8)$$

Substituting these into (5.1.7), we obtain

$$\begin{aligned} \min \quad & \sum_{j=1}^{\ell} \left[\bar{c}_j + \sum_{k=\ell+1}^{n} d_{kj} y_k \right] x_j + \sum_{j=\ell+1}^{n} d_{jo} y_j \\ \text{s.t.} \quad & \sum_{j=1}^{\ell} a_j' x_j = b' \\ & x_j \geq 0 \quad , \quad j = 1,\ldots, \\ & \underline{c}_j \leq y_j \leq \bar{c}_j \quad j = \ell+1,\ldots,n \quad . \end{aligned} \qquad (5.1.9)$$

which is a BLP with special structure. The following theorem characterizes the form of the optimal solution.

Theorem 5.1.1

Let c_j^*, x_j^*, $j = 1,\ldots,n$ be an optimal solution (if it exists at all) of (5.1.4). Then $c_j^* = \underline{c}_j$, $j = 1,\ldots,\ell$ and c_j^* is either $\underline{c}_j$ or $\bar{c}_j$ for $j = \ell+1,\ldots,n$.

Proof

By the fundamental theorem of BLP [13], there is an optimal solution $y^* = (y_{\ell+1}^*,\ldots,y_n^*)$ where y^* is an extreme point of the constraint set $\{(y_{\ell+1},\ldots,y_n) | \underline{c}_j \leq y_j \leq \bar{c}_j,\ j = \ell+1,\ldots,n\}$. ||

We have shown that bilinear programming technique gives a way to solve (5.1.4). This need not, of course, be the best way to solve this class of problems. Typically, the modified version of generalized linear programming algorithm might be able to solve them more efficiently. We will not, however, go into this subject in more detail here.

5.2 Complementary Planning Problems

5.2.1 Problem and Examples

Let us consider the following class of problems

$$\begin{aligned} \text{minimize}\quad & c_1^t x_1 + d_1^t y_1 + c_2^t x_2 + d_2^t y_2 \\ \text{s.t.}\quad & A_1 x_1 + B_1 y_1 \geq b_1 \\ & A_2 x_2 + B_2 y_2 \geq b_2 \\ & x_1 \geq 0 \ ,\ y_1 \geq 0 \ ,\ x_2 \geq 0 \ ,\ y_2 \geq 0 \\ & x_1^t x_2 = 0 \end{aligned} \tag{5.2.1}$$

where $c_1,\ c_2 \varepsilon R^{\ell},\ d_i \varepsilon R^{n_i},\ A_i \varepsilon R^{m_i \times \ell},\ B_i \varepsilon R^{m_i \times n_i},\ b_i \varepsilon R^{m_i},\ i = 1,2$ and x_i, y_i are variable vectors of appropriate dimensions. The last constraint $x_1^t x_2 = 0$ will be called complementary constraints in the sequel.

More general problems with complementary constraints

$$\begin{aligned} \text{minimize}\quad & c_1^t x_1 + c_2^t x_2 + d^t y \\ \text{s.t.}\quad & A_1 x_1 + A_2 x_2 + By \geq b \\ & x_1 \geq 0 \ ,\ x_2 \geq 0 \ ,\ y \geq 0 \\ & x_1^t x_2 = 0 \ . \end{aligned} \tag{5.2.2}$$

has been discussed by Ibaraki [10] [11], who proved the following theorem and proposed an enumeration type of algorithm.

Theorem 5.2.1

If the constraint set of (5.2.2) is bounded, then (5.2.2) has an optimal solution among basic feasible solutions.

Let us first introduce several typical examples of (5.2.1).

a) Complementary Flows in Network

Suppose we want to send two different kinds of flows F_1 and F_2 on a pipe line network. We want to send F_i from source S_i to sink T_i, i = 1,2. Moreover, we assume that two kinds of flows cannot be mixed with each other by some reason or the other. If the capacity of a certain arc consists of many small independent pipes, (more specifically each arc consists of small pipes of 1/2 unit capacity), this problem can be handled by two commodity flow algorithm of Hu [9]. However, if the arc with capacity a_i consists of a single pipe, then we have to have a constraint

$$x_{ij}\ x_{2j} = 0 \quad , \tag{5.2.3}$$

where x_{ij} implies the amount of flow F_i on arc A_j.

Associated with this there are the following three typical problems :

(i) Feasibility problem: Can we send some specified amount of flows F_1 and F_2 without mixing them up?

(ii) Maximum complementary flow problem: What is the maximum sum of flows F_1 and F_2 we can send on the network without mixing two flows?

(iii) Minimum cost complementary flow problem: Find the minimum cost complementary flow satisfying the given flow requirement.

All these problems can be formulated in the framework of (5.2.1).

b) Orthogonal Production Scheduling

It sometimes happens in the optimization of a multi stage production system that the use of certain activities in two consecutive periods

(i) is prohibited (e.g., due to machine maintenance)

(ii) incurs high penalty cost (e.g., hysterisis effects in agricultural production system).

These types of problems can also be put into the form of (5.2.1).

5.2.2 Solution Technique

The most standard way to solve (5.2.1) is to introduce an ℓ-dimensional vector u of 0-1 components and replace the constraints $x_1^t x_2 = 0$ $x_1 \geq 0$, $x_2 \geq 0$ by:

$$x_1 \leq M_o u$$
$$x_2 \leq M_o(e_\ell - u)$$
$$x_1 \geq 0 \quad , \quad x_2 \geq 0 \quad .$$

Here e_ℓ is the ℓ dimensional vector all of whose components are 1's and M_o is a constant satisfying

$$M_o \geq \max \{e_\ell^t x_i \,|\, A_i x_i + B_i y_i \geq b_i,\ x_i \geq 0,\ y_i \geq 0\}$$
$$i = 1,2$$

The equivalence can be seen as follows:

$$u_j = 1 \Rightarrow \{x_{1j} \leq M_o,\ x_{2j} \leq 0,\ x_{1j} \geq 0,\ x_{2j} \geq 0\} \Rightarrow \{x_{2j} = 0,\ x_{1j} \text{ unconstrained}\}$$

$$u_j = 0 \Rightarrow \{x_{1j} \leq 0,\ x_{2j} \leq M,\ x_{ij} \geq 0,\ x_{2j} \geq 0\} \Rightarrow \{x_{1j} = 0,\ x_{2j} \text{ unconstrained}\}$$

Hence (5.2.1) is equivalent to the following mixed 0-1 integer programming problem:

$$\begin{aligned}
\text{maximize}\quad & c_1^t x_1 + d_1^t y_1 + c_2^t x_2 + d_2^t y_2 \quad . \\
\text{s.t.}\quad & A_1 x_1 + B_1 y_1 \geq b_1 \\
& A_2 x_2 + B_2 y_2 \geq b_2 \\
& x_1 - M_o u \quad \leq 0 \\
& x_2 + M_o u \quad \leq M_o e_n \\
& x_1 \geq 0,\quad y_1 \geq 0,\quad x_2 \geq 0,\quad y_2 \geq 0 \\
& u = (u_1, u_2, \ldots, u_\ell) \\
& u_j = 0 \text{ or } 1,\quad j = 1, \ldots, \ell \quad .
\end{aligned} \qquad (5.2.4)$$

This can be solved by a usual branch and bound technique if ℓ is not too large. Instead, we will propose another classical approach, i.e., penalty function approach by putting $x_1^t x_2 = 0$ term into the objective function:

$$\begin{aligned} \text{maximize}\quad & c_1^t x_1 + d_1^t y_1 + c_2^t x_2 + d_2^t y_2 - M x_1^t x_2 \\ \text{s.t.}\quad & A_1 x_1 + B_1 y_1 \geq b_1 \\ & A_2 x_2 + B_2 y_2 \geq b_2 \\ & x_1 \geq 0, \quad y_1 \geq 0, \quad x_2 \geq 0, \quad y_2 \geq 0 \quad . \qquad (5.2.5) \end{aligned}$$

which is a BLP in canonical maximization form.

Theorem 5.2.1 If the constraint set of (5.2.1) is bounded, then there exists a constant M_0 such that (5.2.1) is equivalent to (5.2.5) for $M > M_0$.

Proof

This can be proved by standard technique and will be omitted here.

6. Concluding Remarks

We have picked up several examples of bilinear programs and discussed them in some detail. Some of them are of real practical interest (Chapter 2,3 and Section 5.2) and the others are more of a theoretical nature (Chapter 4 and Section 5.1). The style of presentation is somewhat different for these two groups of examples, but we hope that the readers are more or less convinced of the applicability and importance of bilinear programming through these examples.

The difficulty of nonconvex programs to which bilinear program belongs is the existence of multiple local maxima. The problems treated in Chapter 2 is easier from this viewpoint than those in Chapter 3 and Section 5.2. Also the problems in Chapter 3 are inevitably of high dimensionality and therefore appear to be more difficult than those in Chapter 2. The game theoretic problems of Chapter 4 may appear to be only of theoretical interest to some readers, but the two stage game of Section 4.1 will have more importance in the future as the author has shown in [16]. Also, a constrained bimatrix game of Section 4.2 will have some applications in decisions under multiple objectives.

Bilinear programming is still in its babyhood and a lot of things have to be done if we want to solve a reasonably large real world problem. The efficiency of cutting plane algorithm developed in [13] is not yet authorized by an extensive testing on the computers, though the preliminary results are encouraging. On the other hand, the enumerative approaches of [5] [7] as they stand now appear to be hopelessly expensive for larger problems. Anyway there is a lot of space for improvements in the efficiency of these two groups of algorithms and works in this area will be quite worthwhile.

The last remark is in order: While it is usually difficult to obtain a global optimum, the augmented mountain climbing algorithm of [13] will be useful to obtain a good local optimum without too much expenses.

References

[1] Charnes, A., "Constrained Games and Linear Programming," Proceedings of National Academy of Sciences, 639-649, (1953).

[2] Cottle, R.W. and Dantzig, G.B., "Complementary Pivot Theory of Mathematical Programming," Linear Algebra and Its Applications 1, 103-12.., (1968).

[3] Dantzig, G.B., Linear Programming and Extensions, Princeton University Press, Princeton, N.J., 1963.

[4] ——————, "Solving Two-Move Games with Perfect Information," RAND Report P-1459, Santa Monica, California, 1958.

[5] Falk, J.E., "A Linear Max-Min Problem," Mathematical Programming 5, 169-188, (1973).

[6] Frieze, A.M., "A Bilinear Programming Formulation of the 3-Dimensional Assignment Problem," Mathematical Programming 7, 376-379, (1974).

[7] Gallo, G. and Ülkücü, A., "Bilinear Programming: An Exact Algorithm," Paper presented at 8th International Symposium on Mathematical Programming, Stanford University, California, August, 1973.

[8] Hiller, F.S. and Connors, M.M., "Quadratic Assignment Problem Algorithms and the Location of Indivisible Facilities," Management Science, 13, 52-57, (1966).

[9] Hu, T.C., "Multi Commodity Network Flows," Operations Research, 11, 344-360, (1963).

[10] Ibaraki, T., "Complementary Programming," Operations Research, 19, 1523-1528, (1971).

[11] ——————, "Complementary Programming," Working Paper, Department of Applied Mathematics and Physics, Kyoto University, Kyoto, Japan, 1969.

[12] Keeney, R. and Raiffa, H., Decisions with Multiple Objectives, John Wiley and Sons, Inc., New York, 1976.

[13] Konno, H., "A Cutting Plane Algorithm for Solving Bilinear Programs," IIASA RM-75-61. Laxenburg, Austria, International Institute for Applied Systems Analysis, (to appear in Mathematical Programming).

[14] Konno, H., "Maximization of a Convex Quadratic Functions under Linear Constraints," IIASA RM-75-60. Laxenburg, Austria, International Institute for Applied Systems Analysis, (to appear in Mathematical Programming).

[15] ————, "Finitely Convergent Cutting Plane Algorithm for Bilinear Programs," in preparation.

[16] ————, "Optimum Pricing of Indispensable Raw Material under Monopoly: A Two Stage Game Approach," IIASA-RM Laxenburg, Austria, International Institute for Applied Systems Analysis.

[17] ————, "Bilinear Programming, Part II: Applications of Bilinear Programming," Technical Report 71-10, Department of Operations Research, Stanford University, Stanford, California, 1971.

[18] Lemke, C.E. and Howson J.T., Jr., "Equilibrium Points of Bimatrix Games," SIAM Journal of Applied Mathematics, 12, 413-423, (1964).

[19] Majthay, A. and Whinston, A., "Quasi-Concave Minimization Subject to Linear Constraints," Discrete Mathematics 9, 35-59 (1974).

[20] Mangasarian, O.L., "Equilibrium Points of Bimatrix Games," SIAM Journal of Applied Mathematics, 12, 778-780,(1964).

[21] Nash, J.F., "Non Cooperative Games," Annals of Mathematics, 54, 286-295, (1951)

[22] Pierskalla, W., "The Multi-Dimensional Assignment Problem," Operations Research, 16, 422-431, (1968).

[23] Sarin, R.K., "Iterative Procedures for Evaluation of Multi-Attributed Alternatives," Working Paper 232, Western Management Science Institute, UCLA, Los Angeles, California, 1975.

[24] Scott, A.J., Combinational Programming, Spatial Analysis and Planning, Methuen and Co. Ltd., London, 1971.

[25] von Neumann, J. and Morgenstern, O., Theory of Games and Economic Behavior, Princeton University Press, Princeton, N.J., 1953.

[26] Willoughby, R., (ed.), Sparse Matrix Proceedings, RA1 (11707) IBM Co., Thomas J. Watson Research Center, N.Y., 1969.