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
2. Extended Linear Programs
   2.1 Location-Allocation Problems
   2.2 Applications to Decision Analysis
3. Bilinear Assignment Problems
   3.1 Introduction
   3.2 Multi-Stage Markovian Assignment Problem and Three Dimensional Assignment Problem
   3.3 Reduction of a Sparse Matrix into an Almost Triangular Matrix
4. Game-Theoretic Applications
   4.1 Two Stage Game: A Linear Max-Min Problem
   4.2 An Equilibrium Solution of a Constrained Bimatrix Game
   4.3 Reduction of a Sparse Matrix into an Almost Triangular Matrix
5. Miscellaneous Bilinear Programs
   5.1 Non-Standard Generalized Linear Program
   5.2 Complementary Planning Problems
6. Concluding Remarks
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:

\[
\min \{ c_1^T x_1 + c_2^T x_2 + x_1^T Q x_2 | A_1 x_1 = b_1, x_1 \geq 0, A_2 x_2 = b_2, x_2 \geq 0 \}
\]

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

\[
\min \{ c_1^T x_1 + c_2^T x_2 + x_1^T Q x_2 | A_1 x_1 \geq b_1, x_1 \geq 0, A_2 x_2 \geq b_2, x_2 \geq 0 \}
\]

will be referred to as a bilinear program in canonical minimization form. Bilinear programs in standard and canonical maximization form will be defined analogously. 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-Ülküçü [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 Majthyay and Whinston [19]. Now that there is a workable algorithm, we will pursue further to show the applicability 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.
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]).

\[
\min \{ c^T x | Ax = b, x \geq 0 \} \quad \text{subject to} \quad x \in \mathbb{R}^n
\]

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

\[
C = \{ c \in \mathbb{R}^n | A'c = b', c \geq 0 \}
\]

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

\[
\min \{ c^T x | Ax = b, x \geq 0, A'c = b', c \geq 0 \}
\]

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]:

\[
\min_{c_j, a_j, x_j} \left\{ \sum_{j=1}^{n} c_j x_j \left| \sum_{j=1}^{n} a_j x_j = b, x_j \geq 0, \left( \begin{array}{c} c_j \\ a_j \end{array} \right) \in C_j \subset \mathbb{R}^{m+1} \right. \right. \\
\left. \right. j = 1, ..., n \}
\]

where \( a_j \in \mathbb{R}^m \) and \( C_j \) is a closed convex set, \( j = 1, ..., n \). This program is the source of the famous decomposition algorithm. Here the column vectors \( \left( \begin{array}{c} c_j \\ a_j \end{array} \right) \) as well as \( x_j \)'s are variables and each column \( \left( \begin{array}{c} c_j \\ a_j \end{array} \right) \) is allowed to move in a closed convex set \( C_j \) independently 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):

$$\min \max \{p_1^t x + p_2^t y | B_1 x + B_2 y \geq b\}$$  \hspace{1cm} (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

$$\min \{c_1^t x_1 + c_2^t x_2 | A_1 x_1 = b_1, x_1 \geq 0, A_2 x_2 = b_2, x_2 \geq 0, x_1^t x_2 = 0\}$$  \hspace{1cm} (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):

$$\min \{2c^t x + x^t Q x | Ax = b, x \geq 0\}$$  \hspace{1cm} (1.8)

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

$$\min \{c^t u + c^t v + u^t Q v | Au = b, u \geq 0, Av = b, v \geq 0\}$$  \hspace{1cm} (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 therefore 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](image)

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

\[ b_j = \text{requirement for } M_j \]
\[ u_{ij} = \text{shipment of } M_j \text{ from } P_i \]
\[ a_{ij} = \text{supply of } M_j \text{ at } P_i \]
\[ c_{ij} = \text{unit price of } M_j \text{ at } P_i \]
\[ f_j = \text{unit shipment cost of } M_j \]
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:

\[
\begin{align*}
\sum_{i=1}^{m} u_{ij} &\geq b_j, & j = 1, \ldots, n, \\
0 &\leq u_{ij} \leq a_{ij}, & i = 1, \ldots, m; & j = 1, \ldots, n.
\end{align*}
\]  

(2.1.1)

Total purchasing cost $C_p$ is obviously given by:

\[
C_p = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} u_{ij}
\]  

(2.1.2)

and total transportation cost $C_T$ is given by

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

(2.1.3)

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

i) \textbf{Manhattan Distance}

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

\[
d(P_i, Q) = d_1(P_i, Q) \equiv |p_i - x_o| + |q_i - y_o|
\]  

(2.1.4)

then the total cost $C$ is given by

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

(2.1.5)

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

\[
x_{i1} - x_{i2} = p_i - x_o \quad x_{i1} \geq 0, x_{i2} \geq 0, x_{i1} x_{i2} = 0, i = 1, \ldots, m,
\]

\[
y_{i1} - y_{i2} = q_i - y_o \quad y_{i1} \geq 0, y_{i2} \geq 0, y_{i1} y_{i2} = 0, i = 1, \ldots, m.
\]  

(2.1.6)
the absolute value terms can be written as:

\[ |p_i - x_o| = x_{i1} + x_{i2} \]
\[ |q_i - y_o| = y_{i1} + y_{i2} \]  \hspace{1cm} (2.1.7)

So the problem now is to

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

s.t. \[ \sum_{i=1}^{m} u_{ij} \geq b_j, \quad j = 1, \ldots, n \]
\[ 0 \leq u_{ij} \leq a_{ij} \quad i = 1, \ldots, m, \quad j = 1, \ldots, n \]
\[ x_{i1} - x_{i2} + x_o = p_i, \quad 1 = 1, \ldots, m \]
\[ y_{i1} - y_{i2} + y_o = q_i \]
\[ x_{i1} \geq 0, \quad y_{i1} \geq 0 \quad i = 1, \ldots, m, \quad l = 1, 2, \]
\[ x_{i1} \cdot x_{i2} = 0, \quad y_{i1} \cdot y_{i2} = 0 \quad i = 1, \ldots, m . \]  \hspace{1cm} (2.1.8)

It is straightforward to show that the optimal solution of the associated bilinear program without the orthogonality condition in (2.1.8) automatically satisfy the orthogonality 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.,

\[
d(P_i, Q) = d_2(P_i, Q) = \sqrt{(p_i - x_o)^2 + (q_i - y_o)^2} \]  \hspace{1cm} (2.1.9)

then the problem becomes:
minimize \[ C = \sum_{i=1}^{m} \sum_{j=1}^{n} \left[ c_{ij} + f_j \sqrt{(p_i - x_0)^2 + (q_i y_0)^2} \right] u_{ij} \]
\[ \sum_{i=1}^{m} u_{ij} \geq b_j \quad j = 1, \ldots, n \]
\[ 0 \leq u_{ij} \leq a_{ij} \quad i = 1, \ldots, m \quad j = 1, \ldots, n \]

(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 producing \( L \) different types of commodities \( C_\ell, \ell = 1, \ldots, L \)
(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 \( C_\ell \) 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 \( P_i \)
- \( d_\ell^k \): amount of \( C_\ell \) produced at \( F_k \)
- \( e_{i\ell} \): demand for \( C_\ell \) at \( P_i \)
- \((p_i, q_i)\): location of \( P_i \)
- \((x_k^i, y_k)\): location of \( F_k \)
- \( d(P_i, F_k) \): distance between \( P_i \) and \( F_k \)
- \( f_j \): unit transportation cost of \( M_j \)
- \( g_\ell \): unit transportation cost of \( C_\ell \)
The total cost is now given by

$$C = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{K} c_{ij} u_{ij}^k + \sum_{i=1}^{m} \sum_{k=1}^{K} \left( \sum_{l=1}^{L} g_{i} x_{il}^k + \sum_{j=1}^{n} f_{ij} x_{ij}^k \right) d(P_i, F_k)$$

(2.1.11)

Also $u_{ij}^k$ and $x_{il}^k$ have to satisfy:

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

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

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

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

$$u_{ij}^k \geq 0, \quad x_{il}^k \geq 0.$$  \hspace{1cm} (2.1.12)

Hence now the problem is to minimize (2.1.11) subject to (2.1.12)
which is a BLP if $d(\cdot, \cdot)$ 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 asso­
ciated 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

\[
\begin{align*}
u_i &= \sum_{k=1}^{K} \sum_{j=1}^{N} w_k P_{ij} a_{ij}^k .
\end{align*}
\]  

(2.2.1)

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

It sometimes happens, however, due to the lack of information that the quantities \( w_k, k = 1, \ldots, K \) and \( p_{ij}, i = 1, \ldots, m; j = 1, \ldots, n \) 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

\[
\underline{w}_k \leq w_k \leq \overline{w}_k , \quad k = 1, \ldots, K
\]

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

Similar situation applies as well to the probability measure \( p_{ij} \). Let us suppose here that

\[
P_{ij} \leq p_{ij} \leq \overline{p}_{ij} , \quad i = 1, \ldots, m ; j = 1, \ldots, n
\]

\[
\sum_{j=1}^{n} p_{ij} = 1 \quad i = 1, \ldots, m
\]

where \( p_{ij} \) and \( \overline{p}_{ij} \) 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

\[
W = \{ (w_1, \ldots, w_K) | \underline{w}_k \leq w_k \leq \overline{w}_k \}
\]  

(2.2.2)

and

\[
P_i = \{ (p_{i1}, \ldots, p_{in}) | \underline{p}_{ij} \leq p_{ij} \leq \overline{p}_{ij} \quad j = 1, \ldots, n ; \sum_{j=1}^{n} p_{ij} = 1 \}
\]  

(2.2.3)

which we assume to be nonempty.
Let
\[ u_i = \min \left\{ \sum_{k=1}^{K} \sum_{j=1}^{n} w_k p_{ij} a_{ij}^k \mid w \in W, p_i \in P \right\} \] (2.2.4)
\[ \bar{u}_i = \max \left\{ \sum_{k=1}^{K} \sum_{j=1}^{n} w_k p_{ij} a_{ij}^k \mid w \in W, p_i \in P \right\} \] (2.2.5)

It is obvious that if \( u_r > \bar{u}_s \), then \( A_r \) is preferred to \( A_s \) and \( A_s \) can be eliminated from the candidates of optimal alternatives. Similarly, if
\[ u_r, s = \min \left\{ \sum_{k=1}^{K} \sum_{j=1}^{n} w_k (p_{rj} a_{rj}^k - p_{sj} a_{sj}^k) \mid w \in W, p_r \in P_r, p_s \in P_s \right\} > 0 \] (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 \( i \):
\[ \min \sum_{k=1}^{K} \sum_{j=1}^{n} a_{kj} p_j w_k \]
\[ \text{s.t.} \quad \sum_{j=1}^{n} p_j = 1 , \quad p_j \leq p_j \leq \bar{p}_j , \quad j = 1, \ldots, n ; \]
\[ w_k \leq w_k \leq \bar{w}_k , \quad k = 1, \ldots, K . \] (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 \( w_k \) or \( \bar{w}_k \) for all \( k \). Also, \( \hat{p}_j \) is equal to \( p_j \) 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 $\bar{w}_k$ or $\bar{\bar{w}}_k$. 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:

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

This problem is called a standard linear assignment problem.

Let

\[
\begin{align*}
Z_n = \{(z_{ij}) \in \mathbb{R}^{n^2} & \mid \sum_{j=1}^{n} z_{ij} = 1, \quad i = 1, \ldots, n; \sum_{i=1}^{n} z_{ij} = 1, \\
\quad & \quad j = 1, \ldots, n; \quad z_{ij} \geq 0, \quad i, j = 1, \ldots, n\}
\end{align*}
\]  
(3.1.2)

and

\[
\begin{align*}
Z_n^{I} = \{(z_{ij}) \in \mathbb{R}^{n^2} & \mid (z_{ij}) \in Z_n, \quad z_{ij} = 0 \text{ or } 1, \quad i, j = 1, \ldots, n\}
\end{align*}
\]  
(3.1.3)

It is 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]).

\[
\begin{align*}
\text{maximize} & \quad \sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij} x_{ij} \mid (x_{ij}) \in Z_n
\end{align*}
\]  
(3.1.4)

As a natural extension to the above, the following quadratic

\[
\begin{align*}
\text{maximize} & \quad \sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij} x_{ij} y_{ij} \mid (x_{ij}, y_{ij}) \in Z_n
\end{align*}
\]  
(3.1.5)

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

\[
\text{maximize } \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{r=1}^{n} \sum_{s=1}^{n} a_{rs} x_{ij} y_{rs} \mid (x_{ij}) \in \mathbb{Z}_n^I, (y_{rs}) \in \mathbb{Z}_n^I
\]  

(3.1.5)

has been considered by several authors. Unfortunately, however, \( \mathbb{Z}_n^I \) cannot be replaced by \( \mathbb{Z}_n \) 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:

\[
\text{maximize } \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{r=1}^{n} \sum_{s=1}^{n} a_{ij} x_{ij} y_{rs} \mid (x_{ij}) \in \mathbb{Z}_n, (y_{rs}) \in \mathbb{Z}_n
\]  

(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

\[
\text{maximize } \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{r=1}^{n} \sum_{s=1}^{n} a_{ij} x_{ij} y_{rs} \mid (x_{ij}) \in \mathbb{Z}_n, (y_{rs}) \in \mathbb{Z}_n
\]  

(3.1.7)

**Proof**

(3.1.7) is a bilinear program in standard form and \( \mathbb{Z}_n \) 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 \( \mathbb{Z}_n \). Also, any extreme point of \( \mathbb{Z}_n \) belongs to \( \mathbb{Z}_n^I \) 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 \( J_j \), \( j = 1, \ldots, n \) at the first stage and one and only one of the \( n \) jobs \( J_k \), \( k = 1, \ldots, n \) at the second stage. Here \( \{J_j^1\}^n \) and \( \{J_k^2\}^n \) need not be the same set of jobs. We will assume that the outcome associated with \( M_i + J_j^1 \) at first stage is given by \( p_{ij} \). Due to necessary machine setup to different kind of jobs, the outcome associated with \( M_i + J_k^2 \) 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 \( M_i + J_k^2 \) given \( M_i + J_j^1 \).

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

\[
x_{ij}^\ell = \begin{cases} 
1 & \text{if } M_i \text{ is assigned to } J_j^\ell \text{ at stage } \ell \\
0 & \text{otherwise}
\end{cases}
\] (3.2.1)

Then the two stage optimization problem is formulated as
\[
\max \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} (p_{ij} + p_{ijk})x_{ij}x_{ik} \\
\text{s.t. } (x_{ij}^1) \in \mathbb{Z}^n_{1^J}, \quad (x_{ik}^2) \in \mathbb{Z}_n
\]  \tag{3.2.2}

where \( \mathbb{Z}^n_{1^J} \) is defined by (3.1.3), which is equivalent to

\[
\max \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} (p_{ij} + p_{ijk})x_{ij}^1x_{ik}^2 \\
\text{s.t. } (x_{ij}^1) \in \mathbb{Z}^n_{1^J}, \quad (x_{ik}^2) \in \mathbb{Z}_n
\]  \tag{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 \( M_i + J_k^l \) at stage \( l \) given \( M_i + J_j^{l-1} \) at stage \( l-1 \). Then the problem becomes

\[
\max \sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij}x_{ij} + \sum_{l=2}^{L} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} q_{ijk}^l x_{ij}^{l-1}x_{ik}^l \\
\text{s.t. } (x_{ij}^l) \in \mathbb{Z}^n_{1^J}, \quad l = 1, \ldots, L
\]  \tag{3.2.4}

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

\[
Q_i^l = \begin{bmatrix} q_{i11}^l & q_{i12}^l & \cdots & q_{i1n}^l \\ \vdots & \ddots & \cdots & \vdots \\ q_{in1}^l & \cdots & q_{inn}^l \end{bmatrix} \quad i = 1, \ldots, n; \quad \ell = 1, \ldots, L
\]

\[
Q^l = \begin{bmatrix} Q_1^l \\ \vdots \\ Q_n^l \end{bmatrix}
\]  \tag{3.2.5}  \tag{3.2.6}
For simplicity, let \( L \) be an even integer and let

\[
\begin{pmatrix}
  x^1 \\
  x^2 \\
  \vdots \\
  x^{L-1}
\end{pmatrix}
\quad
\begin{pmatrix}
  v^1 \\
  v^2 \\
  \vdots \\
  v^L
\end{pmatrix}
\]

Then we can rewrite (3.2.4) in a vector form

\[
\begin{align*}
\text{maximize} & \quad p^t x^1 + u^t Q v \\
\text{s.t.} & \quad u^t \in \mathbb{Z}_n^{L/2}, \quad v^t \in \mathbb{Z}_n^{L/2}
\end{align*}
\]  

(3.2.8)

where

\[
Q = \begin{bmatrix}
Q_2 \\
Q_3 \\
\vdots \\
Q_{L-1} \\
Q_L
\end{bmatrix}
\]  

(3.2.9)

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

\[
\begin{align*}
\text{maximize} & \quad p^t x^1 + u^t Q v \\
\text{s.t.} & \quad u^t \in \mathbb{Z}_n^{L/2}, \quad v^t \in \mathbb{Z}_n^{L/2}
\end{align*}
\]  

(3.2.10)
3.2.2 Three Dimensional Assignment Problem

Let there be m research assistants $R_i$, $i = 1, \ldots, m$, n scientists $S_j$, $j = 1, \ldots, n$ and p projects $P_k$, $k = 1, \ldots, p$. 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

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

the problem can be formulated as follows:

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

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
\[ y_{ij} = \begin{cases} 1 & \text{if } R_i \text{ is assigned to } S_j \\ 0 & \text{otherwise} \end{cases} \quad (3.2.13) \]

\[ z_{ik} = \begin{cases} 1 & \text{if } R_i \text{ is assigned to } P_k \\ 0 & \text{otherwise} \end{cases} \quad (3.2.14) \]

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

\[ x_{ijk} = y_{ij} \cdot z_{ik} \cdot \psi_{i,j,k} \quad (3.2.15) \]

as long as the following conditions are satisfied:

\[
\sum_{j=1}^{n} y_{ij} = 1, \quad i = 1, \ldots, m \quad (3.2.16)
\]

Hence (3.2.12) is equivalent to

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

s.t.

\[
\sum_{j=1}^{n} y_{ij} = 1, \quad i = 1, \ldots, m ;
\]

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

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

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

\[ y_{ij}, z_{ik} = 0 \text{ or } 1 . \quad \psi_{i,j,k} \quad (3.2.17) \]
where we used the relationship

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

etc. Let

\[ Y = \{ (y_{ij}) | \sum_{j=1}^{p} y_{ij} = 1, i = 1, \ldots, m; \sum_{i=1}^{m} y_{ij} \leq b_{j}, j = 1, \ldots, n \} \]

\[ Z = \{ (z_{ik}) | \sum_{k=1}^{p} z_{ik} = 1, i = 1, \ldots, m; \sum_{i=1}^{m} z_{ik} \geq c_{k}, k = 1, \ldots, p \} \]

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.

\[
\begin{align*}
\text{max} & \quad \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{p} a_{ijk} y_{ij} z_{ik} \\
\text{s.t.} & \quad (y_{ij}) \in Y, \quad (z_{ik}) \in Z.
\end{align*}
\]

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 \( n \times n \) 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 \( n \times n \) 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

\[
\begin{bmatrix}
1 & 2 & 3 & 4 & 5 \\
\times & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
3 & 4 & 1 & 2 & 5 \\
\times & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
\end{bmatrix}
\]

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 \( A = (a_{ij}) \):

\[
b_{ij} = \begin{cases} 
1 & \text{if } a_{ij} \neq 0 \\
0 & \text{if } a_{ij} = 0 
\end{cases} \quad (3.3.1)
\]

and let

\[
P_n = \{ X = (x_{ij}) \in \mathbb{R}^{n \times n} \mid \sum_{j=1}^{n} x_{ij} = 1, \ i = 1, \ldots, n; \sum_{i=1}^{n} x_{ij} = 1, \ j = 1, \ldots, n \} 
\]

\[
P_I^n = \{ X = (x_{ij}) \in \mathbb{R}^{n \times n} \mid X \in P_n, x_{ij} = 0 \text{ or } 1, \ i, j = 1, \ldots, n \}
\]

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

a) **Arbitrary Permutation of Rows and Columns**

Any combination of row and column permutations of \( B \) can be represented as \( XBY \) where \( X \in P_I^n \) and \( Y \in P_I^n \). 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

\[
\text{minimize } \sum_{j=1}^{n} \sum_{i=j+1}^{n} (XBY)_{ij} \mid X \in P_I^n, Y \in P_I^n \quad (3.3.2)
\]
or

\[
\begin{align*}
\text{minimize} & \quad \sum_{j=1}^{n} \sum_{i=j+1}^{n} \sum_{r=1}^{n} \sum_{s=1}^{n} x_{ir} b_{rs} y_{sj} \\
\text{s.t.} & \quad (x_{ij}) \in Z_{n}^{I}, \quad (y_{ij}) \in Z_{n}^{I}.
\end{align*}
\]

which is equivalent to a standard BLP without integrality conditions

\[
\begin{align*}
\text{minimize} & \quad \sum_{j=1}^{n} \sum_{i=j+1}^{n} \sum_{r=1}^{n} \sum_{s=1}^{n} x_{ir} b_{rs} y_{rj} \\
\text{s.t.} & \quad (x_{ij}) \in Z_{n}, \quad (y_{ij}) \in Z_{n}
\end{align*}
\]

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

\[
\begin{align*}
\text{min} & \quad \ell(X^t, X) = \sum_{j=1}^{n} \sum_{i=j+1}^{n} (X^t B X)_{ij} \\
\text{s.t.} & \quad X \in P_{n}^{I}
\end{align*}
\]

or

\[
\begin{align*}
\text{min} & \quad \ell(X^t, X) = \sum_{j=1}^{n} \sum_{i=j+1}^{n} \sum_{r=1}^{n} \sum_{s=1}^{n} x_{ri} b_{rs} x_{sj} \\
\text{s.t.} & \quad (x_{ij}) \in Z_{n}^{I}
\end{align*}
\]

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
\[ b_{ij}(m) = b_{ij} + m\delta_{ij} \quad , \quad i,j = 1,\ldots,n \]  
where \( m \) is a positive constant and \( \delta_{ij} \) is a standard Kronecker's delta, i.e.,
\[ \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \]

Let us consider the following bilinear program:
\[
\begin{align*}
\min & \quad \ell_{m}(Y,X) = \sum_{j=1}^{n} \sum_{i=j+1}^{n} \sum_{r=1}^{n} \sum_{s=1}^{n} y_{ir} b_{rs}(m) x_{sj} \\
\text{s.t.} & \quad X = (x_{ij}) \in P_{n} \quad , \quad Y = (y_{ij}) \in P_{n} \\
\end{align*}
\]

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 \( m > m_0 \) and \( X^*(m) \) solves (3.3.4) for \( m > m_0 \).

Proof
First note that \( X^*(m) \in P_{n}^{I} \) and \( Y^*(m) \in P_{n}^{I} \). Next, let us prove
\[ z^*_{ij}(m) \equiv \sum_{r=1}^{n} y^*_{ir}(m)x^*_{rj}(m) = \delta_{ij} \quad \forall \ i,j \]  
(3.3.8)
Since \( X^*(m) \in P_{n}^{I} \) and \( Y^*(m) \in P_{n}^{I} \), \( Z^*(m) = (z^*_{ij}(m)) \in P_{n}^{I} \). 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 \( y^*_{lk}(m) = x^*_{lj}(m) \) for \( l = 1,\ldots,n \). This implies that \( y^*_{lk}(m) = x^*_{kj}(m) \) 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
\[
\begin{align*}
\sum_{j=1}^{n} \sum_{i=j+1}^{n} \sum_{r=1}^{n} \sum_{s=1}^{n} y^*_{ir}(m)b_{rs}(m)x^*_{sj}(m) & \geq b_{kk}(m) > m_0 \\
= \sum_{j=1}^{n} \sum_{i=j+1}^{n} \sum_{r=1}^{n} \sum_{s=1}^{n} \delta_{ir} b_{rs}(m)\delta_{sj} \\
\end{align*}
\]
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

\[
\min \{\ell_m(x^t, x) | x \in P_n\} \leq \ell_m(X^*(m)^t, X^*(m))
\]

\[
= \min \{\ell_m(Y, X) | X \in P_n, Y \in P_n\}
\]

and by obvious relation

\[
\min \{\ell_m(x^t, x) | x \in P_n\} \geq \min \{\ell_m(Y, X) | X \in P_n, Y \in P_n\}
\]

we have

\[
\ell_m(X^*(m)^t, X^*(m)) = \min \{\ell_m(x^t, x) | x \in P_n\}
\]

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) P1, P2 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 P1 and P2, respectively. At the first stage of the game, P1 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:

$$X = \{x \in \mathbb{R}^m | A_1 x \leq a_1, x \geq 0\}.$$  \hspace{1cm} (4.1.1)

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

In the second stage of the game, given the full information on the P1's choice of $x \in X$, P2 chooses his action $y$ from the feasible set $Y(x)$ which depends explicitly on $x$. Let us assume here that $Y(x)$ is also defined by the system of linear inequalities:

$$Y(x) = Y \cap B_y(x).$$  \hspace{1cm} (4.1.2)

where

$$Y = \{y \in \mathbb{R}^n | A_2 y \leq a_2, y \geq 0\}.$$ \hspace{1cm} (4.1.3)

$$B_y(x) = \{y \in \mathbb{R}^n | B_2 y \leq b - B_1 x, y \geq 0\}$$

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

We will assume here that $Y(x)$ is nonempty and compact for all
As to the payoff structure of this game, we will assume that \( P_1 \) has to pay
\[
f(x, y) = p^tx + p^ty
\]
to \( P_2 \) when \( P_1 \) and \( P_2 \) choose \( x \in X \) and \( y \in Y(x) \), respectively, where \( p_1 \in \mathbb{R}^m \), \( p_2 \in \mathbb{R}^n \) are given vectors.

Given \( x \in X \), \( P_2 \) naturally wants to maximize \( f(x, y) \) over \( y \in Y(x) \). Hence his problem is to
\[
\text{maximize } \{p_2^ty | y \in Y(x)\} \quad (4.1.5)
\]
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 \( x \in X \). The problem for \( P_1 \) is now to
\[
\text{minimize } \{p_1^tx + p_2^ty^*(x) | x \in X\} \quad (4.1.6)
\]
or equivalently,
\[
\min_{x \in X} \max_{y \in Y} \{p_1^tx + p_2^ty | B_1x + B_2y \leq b\} \quad (4.1.7)
\]
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
\[
\max \{p_2^ty | B_2y \leq b - B_1x, A_2y \leq a_2, y \geq 0\} \quad (4\text{.1.8})
\]
Taking the dual of this linear program:
\[
\min \{(b - B_1x)^t z_1 + a_2^t z_2 | B_2^t z_1 + A_2^t z_2 \geq p_2, z_1 \geq 0, z_2 \geq 0\}
\]
we get a bilinear program which is equivalent to (4.1.7)
\[
\begin{align*}
\min & \quad p_1^tx + (b - B_1x)^t z_1 + a_2^t z_2 \\
\text{s.t.} & \quad A_1x \leq a_1, \quad x \geq 0, \\
& \quad B_2^t z_1 + A_2^t z_2 \geq p_2, \quad z_1 \geq 0, \quad z_2 \geq 0.
\end{align*}
\]
Let us consider now the linear min-max problem associated with
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

\[
X = \{(x_1, \ldots, x_m) \mid \sum_{i=1}^{m} x_i = 1 \quad , \quad x_i \geq 0 \quad , \quad i = 1, \ldots, m\}
\]

\[
Y = \{(y_1, \ldots, y_n) \mid \sum_{j=1}^{n} y_j = 1 \quad , \quad y_j \geq 0 \quad , \quad j = 1, \ldots, n\}
\]

and where the payoff is given by \( x^tAy \) where \( A \) is a given \( m \times n \) matrix the famous min-max theorem [25] holds:

\[
\max_{y \in Y} \min_{x \in X} x^tAy = \min_{x \in X} \max_{y \in Y} x^tAy.
\]

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

Let

\[
B_y(x) = \{y \in \mathbb{R}^n \mid B_2y \leq b - B_1x\}
\]

\[
B_x(y) = \{x \in \mathbb{R}^m \mid B_1x \leq b - B_2y\}.
\]

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

**Theorem 4.1.1**

Under the assumption we made,

\[
\max_{y \in Y} \min_{x \in X} \{p_1^tx + p_2^ty \mid B_1x + B_2y \leq b\} \geq \min_{x \in X} \max_{y \in Y} \{p_1^tx + p_2^ty \mid B_1x + B_2y \leq b\}.
\]
Proof

$$\max \min \{ p^t_1 x + p^t_2 y | B_1 x + B_2 y \leq b \}$$

$$\leq \max \{ p^t_2 y + \min \{ p^t_1 x | B_1 x \leq b - B_2 y \} \}$$

$$\geq \max \{ p^t_2 y + \min \{ p^t_1 x \} \}$$

$$\geq \min \{ p^t_1 x + \max \{ p^t_2 y \} \}$$

$$\leq \min \{ p^t_1 x + \max \{ p^t_2 y | B_2 y \leq b - B_1 x \} \}$$

$$= \min \max \{ p^t_1 x + p^t_2 y | B_1 x + B_2 y \leq b \} \ .$$

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:

$$\max p^t_2 y + (b - B_2 y)^t z_1 + a^t z_2$$

s.t. \(A_2 y \leq a_2\), \(y \geq 0\) \ .

$$B_1^t z_1 + A_1^t z_2 \geq p_1\), \(z_1 \geq 0\), \(z_2 \geq 0\) \ .

(4.1.12)

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

$$\min \{ f(u, z_1, z_2) | u \in U, (z_1, z_2) \in Z \} \ \ \ \ \ \ \ \ \ (4.1.13)$$

$$\max \{ f(u, z_1, z_2) | u \in U, (z_1, z_2) \in Z \} \ \ \ \ \ \ \ \ \ (4.1.14)$$

where

\(f(u, z_1, z_2) = p^t u + (b - Bu)^t z_1 + a^t z_2\) \ .

\(U = \{ u | Au \leq a, u \geq 0 \}\)

\(Z = \{ (z_1, z_2) | B^t z_1 + A^t z_2 \geq p, z_1 \geq 0, z_2 \geq 0 \} \ .\)
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 P₁, P₂ and let S = \{s₁, s₂, ..., sₘ\} and T = \{t₁, t₂, ..., tₙ\} be the sets of actions available to P₁ and P₂, respectively. Assume that the payoffs to P₁ and P₂ when P₁ chooses sᵢ and P₂ chooses tⱼ are given by aᵢⱼ and bᵢⱼ, respectively. Given two matrices A = (aᵢⱼ), B = (bᵢⱼ), each player chooses the mixed strategies (or probability measure) on S and T, i.e., P₁ chooses \( x ∈ X₀ \) and P₂ chooses \( y ∈ Y₀ \) where

\[
X₀ = \{ x ∈ \mathbb{R}^m | e_m^t x = 1, \ x ≥ 0 \} \\
Y₀ = \{ y ∈ \mathbb{R}^n | e_n^t y = 1, \ y ≥ 0 \}
\]

where \( e_m \) and \( e_n \) are m and n dimensional vectors all of whose components are ones. Then the expected payoffs to P₁ and P₂ are given by \( x^t Ay \) and \( x^t By \), respectively. (Note that the two players P₁ and P₂ choose \( x \) and \( y \) simultaneously). Let us denote this game as \( Γ(A,X₀; B,Y₀) \).

**Definition 4.2.1**

A pair of mixed strategies \( (\tilde{x}, \tilde{y}) ∈ X₀ × Y₀ \) is a Nash equilibrium point of a bimatrix game \( Γ(A,X₀; B,Y₀) \) if

\[
\tilde{x}^t A\tilde{y} = \max \{ x^t A\tilde{y} | x ∈ X₀ \} \quad (4.2.3) \\
\tilde{x}^t B\tilde{y} = \max \{ \tilde{x}^t B\tilde{y} | y ∈ Y₀ \} \quad (4.2.4)
\]

The Nash equilibrium point implies that as long as P₁ sticks to \( \tilde{x} \), then \( \tilde{y} \) is the optimal mixed strategies for P₂ and vice versa. It has been shown in [2] that the problem of obtaining a Nash equilibrium point of \( (A,X₀; B,Y₀) \) can be reduced to a linear complementarity problem (LCP):
Find: \( z \in \mathbb{R}^{m+n} \) 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 ingenious 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 \( x \) and \( y \) from more general constraint sets \( X \) and \( Y \), respectively. We will call \( X \) and \( Y \) admissible sets and \( x \in X \), \( y \in Y \) admissible strategies.

**Definition 4.2.2**

A pair of admissible strategies \((x, y) \in X \times Y\) is called a Nash equilibrium point of \( \Gamma(A, X; B, Y) \) if
\[
    \begin{align*}
    \tilde{x}^tA\tilde{y} &= \max \{x^tA\tilde{y} | x \in X\} \quad (4.2.5) \\
    \tilde{x}^tB\tilde{y} &= \max \{\tilde{x}^tA\tilde{y} | y \in Y\} \quad (4.2.6)
    \end{align*}
\]

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 \( \Gamma(A, X; B, Y) \) where
\[
    X_1 = \{x \in \mathbb{R}^m | P_1x = P_1, \ x \geq 0\} \quad (4.2.7)
\]
\[
    Y_1 = \{y \in \mathbb{R}^n | P_2y = P_2, \ y \geq 0\} \quad (4.2.8)
\]
where \( P_i \in \mathbb{R}^{i \times m_i} \), \( P_i \in \mathbb{R}^{i \times l_i} \), \( i = 1,2 \).
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**

\((x, y) \in X \times Y\) 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 \( \xi \in \mathbb{R} \) and \( \eta \in \mathbb{R} \).

\[
\begin{align*}
\dot{x}^A y - p_1^t u &= 0 \\
A y - p_1^t u &\leq 0 \\
p_1 \dot{x} &= p_1 \\
x &\geq 0 \quad (4.2.8) \\
\dot{y} &\geq 0 \quad (4.2.9)
\end{align*}
\]

**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)

\[
\begin{align*}
\text{maximize } \phi(x, y, u, v) \\
&= x^t (A+B)y - p_1^t u - p_2^t v \\
\text{s.t. } &B^t x - p_2^t v \leq 0 \\
&\quad p_1 x = p_1 \\
&\quad A y - p_1^t u \leq 0 \\
&\quad p_2 y = p_2 \\
&\quad x \geq 0 \quad , \quad y \geq 0 . \quad (4.2.10)
\end{align*}
\]
Theorem 4.2.3

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 \((\bar{x}, \bar{y})\) is an optimal solution of (4.2.10). Also \(\phi(x, y, u, v) = 0\) at the optimum.

Proof

Let \((\bar{x}, \bar{y})\) be an equilibrium point of \(\Gamma(A, X_1; B, Y_1)\). Then by (4.2.8) and (4.2.9), \((\bar{x}, \bar{y})\) together with \((\bar{u}, \bar{v})\) satisfy the constraints of (4.2.10), i.e., \((\bar{x}, \bar{y}, \bar{u}, \bar{v})\) is a feasible solution of (4.2.10). Also \(\phi(\bar{x}, \bar{y}, \bar{u}, \bar{v}) = 0\). However, by premultiplying \(x^t, y^t\) to the first and the third inequalities of (4.2.10) and using the facts that \(P_1 x = P_1\) and \(P_2 y = P_2\), we obtain the inequality.

\[
0 \geq x^t A y - x^t P_1 u = x^t A y - P_1^t u
\]
\[
0 \geq x^t B y - v^t Q y = x^t B y - P_2^t v
\]

and hence \(\phi(x, y, u, v) \leq 0\) for all feasible solution of (4.2.10), whence \((\bar{x}, \bar{y}, \bar{u}, \bar{v})\) 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

\[
(x^*)^t A y^* - P_1^t u^* = (x^*)^t B y^* - P_2^t v^* = 0
\]

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:

\[
\begin{align*}
\min & \sum_{j=1}^{n} c_j x_j \\
\text{s.t.} & \sum_{j=1}^{n} a_j x_j = b , \\
& x_j \geq 0 , \quad \left( \begin{array}{c} c_j \\ a_j \end{array} \right) \in C_j .
\end{align*}
\]

(5.1.1)

where \( a_j \in \mathbb{R}^m \), \( c_j \in \mathbb{R}^l \) and \( C_j \subseteq \mathbb{R}^{m+l} \) is a compact convex set \( j = 1, \ldots, n \) and maximization is with respect to \( \left( \begin{array}{c} c_j \\ a_j \end{array} \right) \) as well as \( x_j \). The GLP algorithm by Dantzig and Wolfe proceeds roughly as follows:

Let \( \left( \begin{array}{c} c_j^l \\ a_j^l \end{array} \right) \in C_j^l \), \( l = 1, \ldots, l_j \), \( j = 1, \ldots, n \) be given.

Then we will solve the linear program:

\[
\begin{align*}
\min & \sum_{j=1}^{n} \sum_{l=1}^{l_j} c_j^l x_j^l \\
\text{s.t.} & \sum_{j=1}^{n} \sum_{l=1}^{l_j} a_j^l x_j^l = b , \\
& x_j^l \geq 0 , \quad l = 1, \ldots, l_j ; \quad j = 1, \ldots, n
\end{align*}
\]

(5.1.2)

and let \( \bar{\pi} \in \mathbb{R}^m \) be an optimal multiplier vector for this linear program.

If

\[
c_j - \bar{\pi} a_j \geq 0 , \quad \left( \begin{array}{c} c_j \\ a_j \end{array} \right) \in C_j ; \quad j = 1, \ldots, n .
\]

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

\[ c_j - \pi a_j > 0, \]

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

\[
\min \{ c_j - \pi a_j \mid \begin{pmatrix} c_j \\ a_j \end{pmatrix} \in C_j \}, \quad j = 1, \ldots, n . \tag{5.1.3}
\]

Let \( \begin{pmatrix} c_j^* \\ a_j^* \end{pmatrix} \) be its optimal solution. If \( c_j^* - \pi a_j^* < 0 \), 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.,

\[
\min \sum_{j=1}^{n} c_j x_j \\
\text{s.t.} \quad \sum_{j=1}^{n} a_j x_j = b \\
x_j \geq 0 , \quad j = 1, \ldots, \ell ; \\
x_j \leq 0 , \quad j = \ell+1, \ldots, n ; \tag{5.1.4}
\]

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

\[ x_j = x_{j1} - x_{j2} , \quad x_{j1} \geq 0 , \quad x_{j2} \geq 0 , \quad j = \ell+1, \ldots, n . \]
then the problem is

$$\begin{align*}
\min & \sum_{j=1}^{\ell} c_j x_j + \sum_{j=\ell+1}^{n} c_{j1} x_{j1} - \sum_{j=\ell+1}^{n} c_{j2} x_{j2} \\
\text{s.t.} & \sum_{j=1}^{\ell} a_j x_j + \sum_{j=\ell+1}^{n} a_{j1} x_{j1} - \sum_{j=\ell+1}^{n} a_{j2} x_{j2} = b \\
x_j & \geq 0 , \quad j = 1, \ldots, \ell \\
x_{j1}, x_{j2} & \geq 0 , \quad j = \ell+1, \ldots, n \\
(c_j) & \in C_j, \ j = 1, \ldots, \ell. \\
(a_j) & \in C_j, \ j = \ell+1, \ldots, n.
\end{align*}$$

(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

$$\begin{align*}
\min & \sum_{j=1}^{\ell} c_j x_j + \sum_{j=\ell+1}^{n} c_{j1} x_{j1} \\
\text{s.t.} & \sum_{j=1}^{\ell} a_j x_j + \sum_{j=\ell+1}^{n} a_{j1} x_{j1} = b \\
x_j & \geq 0 , \quad j = 1, \ldots, \ell \\
c_j & \leq c_j \leq \bar{c}_j , \quad j = 1, \ldots, n.
\end{align*}$$

(5.1.6)

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

s.t. \[ \sum_{j=1}^{\ell} a_j x_j + \sum_{j=\ell+1}^{n} a_j x_j = b \]
\[ x_j \geq 0, \quad j = 1, \ldots, \ell ; \]
\[ c_j \leq y_j \leq \bar{c}_j, \quad j = \ell+1, \ldots, n . \] (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

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

Substituting these into (5.1.7), we obtain

\[ \min \sum_{j=1}^{\ell} \left[ \bar{c}_j + \sum_{k=\ell+1}^{n} d_{kj} y_k \right] x_j + \sum_{j=\ell+1}^{n} d_{jo} y_j \]

s.t. \[ \sum_{j=1}^{\ell} a_j x_j = b' \]
\[ x_j \geq 0, \quad j = 1, \ldots, \]
\[ c_j \leq y_j \leq \bar{c}_j, \quad j = \ell+1, \ldots, n . \] (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 \( c_j^* = c_j^* \), \( j = 1, \ldots, \ell \) and \( c_j^* \) is either \( c_j \) 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_{l+1}, \ldots, y_n) | c_j \leq y_j \leq \bar{c}_j, j = l+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

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

(5.2.1)

where \( c_1, c_2 \in \mathbb{R}^n_i, d_1 \in \mathbb{R}^m_i, A_i \in \mathbb{R}^{m_i \times n_i}, B_i \in \mathbb{R}^{m_i \times n_i}, b_i \in \mathbb{R}^{m_i}, i = 1, 2 \)

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

\[
\begin{align*}
\text{minimize} & \quad c_1^t x_1 + c_2^t x_2 + d^t y \\
\text{s.t.} & \quad A_1 x_1 + A_2 x_2 + B y \geq b \\
& \quad x_1 \geq 0, \quad x_2 \geq 0, \quad y \geq 0 \\
& \quad x_1^t x_2 = 0.
\end{align*}
\]

(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 pipeline 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

$$x_{ij} x_{2j} = 0,$$  \hspace{1cm} (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., hysteresis 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 \( \ell \)-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:

\[
\begin{align*}
x_1 &\leq M_0 u \\
x_2 &\leq M_0 (e^\ell - u) \\
x_1 &\geq 0 \, , \, x_2 \geq 0 
\end{align*}
\]

Here \( e^\ell \) is the \( \ell \) dimensional vector all of whose components are 1's and \( M_0 \) is a constant satisfying

\[ M_0 \geq \max \{ e^\ell x_i | A_i x_i + B_i y_i \geq b_i , x_i \geq 0 , y_i \geq 0 \} \]

\[ i = 1, 2 \]

The equivalence can be seen as follows:

\[ u_j = 1 \rightarrow \{ x_{1j} \leq M_0 , \, x_{2j} \leq 0 , \, x_{1j} \geq 0 , \, x_{2j} \geq 0 \} \rightarrow \{ x_{2j} = 0 , \, x_{1j} \text{ unconstrained} \} \]

\[ u_j = 0 \rightarrow \{ x_{1j} \leq 0 , \, x_{2j} \leq M , \, x_{1j} \geq 0 , \, x_{2j} \geq 0 \} \rightarrow \{ x_{1j} = 0 , \, x_{2j} \text{ unconstrained} \} \]

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

\[
\begin{align*}
\text{maximize} & \quad c_1^t x_1 + d_1^t y_1 + c_2^t x_2 + d_2^t y_2 \\
\text{s.t.} & \quad A_1 x_1 + B_1 y_1 \geq b_1 \\
& \quad A_2 x_2 + B_2 y_2 \geq b_2 \\
& \quad x_1 - M_0 u \leq 0 \\
& \quad x_2 + M_0 u \leq M e^n \\
& \quad x_1 \geq 0 , \, y_1 \geq 0 , \, x_2 \geq 0 , \, y_2 \geq 0 \\
& \quad u = (u_1, u_2, \ldots, u_\ell) \\
& \quad u_j = 0 \text{ or } 1 \, , \, j = 1, \ldots, \ell 
\end{align*}
\]
This can be solved by a usual branch and bound technique if λ 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:

$$\text{maximize } c_1^t x_1 + d_1^t y_1 + c_2^t x_2 + d_2^t y_2 - Mx_1^t x_2$$

s.t.  
$$A_1 x_1 + B_1 y_1 \geq b_1$$  
$$A_2 x_2 + B_2 y_2 \geq b_2$$  
$$x_1 \geq 0, \ y_1 \geq 0, \ x_2 \geq 0, \ y_2 \geq 0 \ .$$

(5.2.5)

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


