Efficient Algorithms for Solving Rank Two and Rank Three Bilinear Programming Problems*

YASUTOSHI YAJIMA
Department of Industrial Engineering and Management, Tokyo Institute of Technology, Japan

and

HIROSHI KONNO
Institute of Human and Social Sciences, Tokyo Institute of Technology, Japan

(Received: 25 July 1990; accepted: 18 December 1990)

Abstract. Finitely convergent algorithms for solving rank two and three bilinear programming problems are proposed. A rank k bilinear programming problem is a nonconvex quadratic programming problem with the following structure:

$$\text{minimize}\left\{c_0^t x + d_0^t y + \sum_{j=1}^{k} c_j^t x \cdot d_j^t y \,\middle|\, x \in X,\ y \in Y\right\},$$

where $X \subset R^{n_1}$ and $Y \subset R^{n_2}$ are non-empty and bounded polytopes. We show that a variant of parametric simplex algorithm can solve large scale rank two bilinear programming problems efficiently. Also, we show that a cutting-cake algorithm, a more elaborate variant of parametric simplex algorithm can solve medium scale rank three problems.

Key words. Bilinear programming, parametric simplex algorithm, global minimization, nonconvex quadratic programming problem.

1. Introduction

In this paper, we will propose rigorous and efficient algorithms for solving rank two and rank three bilinear programming problems. A bilinear programming problem (BLP) is a special type of nonconvex quadratic programming problem:

$$\left|\begin{array}{ll} \text{minimize} & f(x, y) = c_0^t x + d_0^t y + x^t C y \\ \text{subject to} & A_1 x = b_1\,, \quad x \geq 0\,, \\ & A_2 y = b_2\,, \quad y \geq 0\,, \end{array}\right. \tag{1.1}$$

where $c_0 \in R^{n_1}$, $d_0 \in R^{n_2}$, $C \in R^{n_1 \times n_2}$, $A_i \in R^{m_i \times n_i}$, $b_i \in R^{m_i}$, $(i = 1, 2)$. Let us denote

$$X = \{x \in R^{n_1} \mid A_1 x = b_1, x \geq 0\}\,, \tag{1.2}$$

$$Y = \{y \in R^{n_2} \mid A_2 y = b_2, y \geq 0\}\,. \tag{1.3}$$

*This research was supported in part by Grant-in-Aid for Scientific Research of the Ministry of Education, Science and Culture, Grant No. 63490010.

© 1991 *Kluwer Academic Publishers. Printed in the Netherlands.*

We will assume throughout this paper that

$$X \text{ and } y \text{ are nonempty and bounded}. \tag{1.4}$$

When rank C is k, (1.1) will be called a rank k bilinear programming problem. It is shown in [9] that a rank k bilinear programming problem can be put into its canonical form:

$$\left|\begin{array}{ll} \text{minimize} & f(x, y) = c_0^t x + d_0^t y + \sum_{j=1}^{k} c_j^t x \cdot d_j^t y \\ \text{subject to} & x \in X, \quad y \in Y, \end{array}\right. \tag{1.5}$$

where both $\{c_1, c_2, \ldots, c_k\}$ and $\{d_1, d_2, \ldots, d_k\}$ are linearly independent sets of vectors.

BLP's are related to several important classes of nonconvex minimization programming problems [5, 6]. First of all, a concave quadratic programming problem:

$$\left|\begin{array}{ll} \text{minimize} & q^t z + \frac{1}{2} z^t Q z \\ \text{subject to} & Az = b, \quad z \geq 0, \end{array}\right. \tag{1.6}$$

where Q is a symmetric negative semi-definite matrix, is equivalent to a bilinear programming problem:

$$\left|\begin{array}{ll} \text{minimize} & q^t x + q^t y + x^t Q y \\ \text{subject to} & Ax = b, \quad x \geq 0, \\ & Ay = b, \quad y \geq 0 \end{array}\right. \tag{1.7}$$

as stated in the following theorem [8]:

THEOREM 1.1. *Let (x^*, y^*) be an optimal solution of (1.7). Then both x^* and y^* are optimal solution of (1.6). Conversely, if z^* is an optimal solution of (1.6), then $(x, y) = (z^*, z^*)$ is an optimal solution of (1.7).*

It is proved in [6] that the feasibility problem of 0–1 integer programming problems can be reformulated as concave quadratic programming problems (1.5), which implies that 0–1 integer linear programming problems can be converted to a sequence of bilinear programming problems.

Second important class of problems are linear min-max problems [2]:

$$\max_y \min_x \{p^t x + q^t y \mid F_1 x + F_2 y = f, \quad x \geq 0, \quad y \geq 0\}. \tag{1.8}$$

By taking the partial dual of (1.1) with respect to y, we obtain a linear min-max problem:

$$\min_x \max_u \{c_0^t x + b_2^t u \mid A_1 x = b_1, \quad A_2^t u - C^t x \geq d_0, \quad x \geq 0\}. \tag{1.9}$$

Conversely, a linear min-max problem can be converted to a bilinear programming problem.

It is shown in [6, 17] that a number of important real world problems can be formulated as bilinear programming problems.

Motivated by theoretical and practical importance, many researchers tried to construct efficient algorithms for bilinear programming problems. Among them are cutting plane algorithms [7, 12, 13, 15, 18], successive under-estimation methods [3], branch and bound algorithms [4, 14, 15, 16]. All of these algorithms exploit the important property of BLP's stated in the following theorem [7].

THEOREM 1.2. *Under assumption (1.4), a bilinear programming problem has an optimal solution (x^*, y^*) such that x^* and y^* are extreme points of X and Y, respectively.*

Unfortunately, however some of these algorithms, though reasonably efficient, may not be finitely convergent. Finitely convergent algorithms, on the other hand are usually inefficient. In particular, some finite algorithms are not more efficient than the total enumeration of extreme points of X and Y.

Recently, the authors [9] proposed a finite and efficient algorithm for obtaining a global minimum of rank two bilinear programming problems. This algorithm uses a variant of parametric simplex method. According to our numerical experiments [9], totally dense problems of the size up to $(m_1, m_2, n_1, n_2) = (100, 100, 180, 180)$ can be solved within 10 minutes on SUN4/280 workstation. Computation time would have been less if the problems are sparse.

In Section 2, we will discuss some improvements on the rank two bilinear programming algorithm proposed in [9]. It turns out that this improvement is quite substantial. In fact, we can now solve general rank two bilinear programming problems in much the same computational time as that needed to solve three linear programs with the same constraints.

In Section 3, we will propose a "cutting cake algorithm" for solving rank three problems. This algorithm partitions the polytope of the three dimensional parameter space into pieces of cheese cake shaped subregions. The algorithm extensively uses parametric linear programming technique. It is exact, finite and is demonstrated to be reasonably efficient.

Finally, in Section 4, we will discuss future directions of research.

2. A Fast Algorithm for General Rank Two Bilinear Programs

2.1. PARAMETRIC REPRESENTATION

Let us consider the general rank two bilinear programming problem:

$$\left|\begin{array}{ll} \text{minimize} & G_2(x, y) = c_0^t x + d_0^t y + c_1^t x \cdot d_1^t y + c_2^t x \cdot d_2^t y \\ \text{subject to} & x \in X, \quad y \in Y, \end{array}\right. \tag{2.1}$$

in which both $\{c_0, c_1, c_2\}$ and $\{d_0, d_1, d_2\}$ are linearly independent.

Remark: If c_0 is dependent on c_1, c_2 or d_0 is dependent on d_1, d_2, then the problem can be solved by a single parameter simplex algorithm proposed in [9]. The algorithm is so efficient that further improvement appears impossible.

To solve (2.1), we first introduce a pair of auxiliary variables:

$$\begin{cases} \xi = c_1^t x \,, \\ \eta = d_2^t y \,, \end{cases} \tag{2.2}$$

and put (2.1) into its parametric representation:

$$\left|\begin{array}{ll} \text{minimize} & g_2(x, y; \xi, \eta) = c_0^t x + d_0^t y + \xi d_1^t y + \eta c_2^t x \\ \text{subject to} & x \in X\,, \quad y \in Y\,, \\ & c_1^t x = \xi\,, \quad d_2^t y = \eta\,, \\ & \xi_{\min} \leq \xi \leq \xi_{\max}\,, \quad \eta_{\min} \leq \eta \leq \eta_{\max}\,, \end{array}\right. \tag{2.3}$$

where

$$\begin{cases} \xi_{\min} = \min\{c_1^t x \mid A_1 x = b_1\,, \quad x \geq 0\}\,, \\ \xi_{\max} = \max\{c_1^t x \mid A_1 x = b_1\,, \quad x \geq 0\}\,, \\ \eta_{\min} = \min\{d_2^t y \mid A_2 y = b_2\,, \quad y \geq 0\}\,, \\ \eta_{\max} = \max\{d_2^t y \mid A_2 y = b_2\,, \quad y \geq 0\}\,. \end{cases} \tag{2.4}$$

Let

$$\Pi = \{(\xi, \eta) \mid \xi_{\min} \leq \xi \leq \xi_{\max}\,, \quad \eta_{\min} \leq \eta \leq \eta_{\max}\}\,. \tag{2.5}$$

Let us define a subproblem:

$$P(\xi, \eta)\left|\begin{array}{ll} \text{minimize} & g_2(x, y; \xi, \eta) = c_0^t x + d_0^t y + \xi d_1^t y + \eta c_2^t x \\ \text{subject to} & x \in X\,, \quad y \in Y\,, \\ & c_1^t x = \xi\,, \quad d_2^t y = \eta\,, \end{array}\right. \tag{2.6}$$

where $(\xi, \eta) \in \Pi$. Let

$$h(\xi, \eta) = \min\{g_2(x, y; \xi, \eta) \mid x \in X, y \in Y, c_1^t x = \xi, d_2^t y = \eta\}\,.$$

To solve (2.3), it sufficies to find the global minimum point (ξ^*, η^*) of $h(\xi, \eta)$ over $(\xi, \eta) \in \Pi$. The global minimum (x^*, y^*) of (2.1) can be obtained by solving $P(\xi^*, \eta^*)$.

Let us proceed to the algorithm for obtaining (ξ^*, η^*).

2.2. PARTITIONING OF THE TWO DIMENSIONAL PARAMETER SPACE

The crucial observation is that $P(\xi, \eta)$ can be decomposed into two subproblems:

$$P_X(\xi, \eta)\left|\begin{array}{ll} \text{minimize} & (c_0 + \eta c_2)^t x \\ \text{subject to} & x \in X\,, \quad c_1^t x = \xi\,, \end{array}\right. \tag{2.7}$$

$$P_Y(\xi, \eta)\left|\begin{array}{ll} \text{minimize} & (d_0 + \xi d_1)^t y \\ \text{subject to} & y \in Y\,, \quad d_2^t y = \eta\,, \end{array}\right. \tag{2.8}$$

which can be solved separately.

Let us define two functions on Π:

$$h_X(\xi,\eta)=\min\{(c_0+\eta c_2)^t x \mid x\in X, \quad c_1^t x=\xi\},$$
$$h_Y(\xi,\eta)=\min\{(d_0+\xi d_1)^t y \mid y\in Y, \quad d_2^t y=\eta\}.$$

Then

$$h(\xi,\eta)=h_X(\xi,\eta)+h_Y(\xi,\eta).$$

Let $(\xi_0,\eta_0)\in\Pi$ and consider a linear program:

$$P_X(\xi_0,\eta_0)\left|\begin{array}{ll}\text{minimize} & (c_0+\eta_0 c_2)^t x\\ \text{subject to} & A_1 x=b_1, \quad x\geqslant 0,\\ & c_1^t x=\xi_0.\end{array}\right. \tag{2.9}$$

Let B be an optimal basis of $P_X(\xi_0,\eta_0)$ and N be the associated nonbasic matrix.

Then we have

$$\begin{cases}\bar c_{0N}+\eta_0\bar c_{2N}\geqslant 0,\\ B^{-1}\begin{pmatrix}b_1\\ \xi_0\end{pmatrix}\geqslant 0.\end{cases} \tag{2.10}$$

where

$$\bar c_{0N}^t=c_{0N}^t-c_{0B}^t B^{-1}N,$$
$$\bar c_{2N}^t=c_{2N}^t-c_{2B}^t B^{-1}N.$$

Also, B remains optimal for $P_X(\xi,\eta)$ as long as

$$\begin{cases}\bar c_{0N}+\eta\bar c_{2N}\geqslant 0\\ B^{-1}\begin{pmatrix}b_1\\ \xi\end{pmatrix}\geqslant 0,\end{cases} \tag{2.11}$$

from which we obtain a rectangle (Figure 1)

$$R(\underline{\xi},\bar\xi;\underline{\eta},\bar\eta)\equiv\{(\xi,\eta)\mid \underline{\xi}\leqslant\xi\leqslant\bar\xi, \quad \underline{\eta}\leqslant\eta\leqslant\bar\eta\}.$$

B is optimal for $P_X(\xi,\eta)$ for all $(\xi,\eta)\in R(\underline{\xi},\bar\xi;\underline{\eta},\bar\eta)$. Also note that $h_X(\xi,\eta)$ is a bilinear function of (ξ,η) on $R(\underline{\xi},\bar\xi;\underline{\eta},\bar\eta)$. When (ξ,η_0) reaches the edge of the rectangle $R(\underline{\xi},\bar\xi;\underline{\eta},\bar\eta)$, say $(\bar\xi,\eta_0)$, then an alternative optimal basis will be generated by applying a dual simplex pivot. Also when (ξ_0,η) reaches $(\xi_0,\bar\eta)$ then an alternative basis will be generated by applying a primal simplex pivot. Thus we obtain a partition of the rectangle Π into a finite number of subrectangles (See Figure 2) by applying a sequence of primal and/or dual simplex pivots by barring degeneracy appropriately (For details, see [10] where we developed a criss-cross method to get around degeneracy.).

Let $\mathcal{R}_X$ be the collection of subrectangles covering Π.

Analogously, we can obtain another collection $\mathcal{R}_Y$ of subrectangles covering Π by applying the same procedure to $P_Y(\xi,\eta)$.

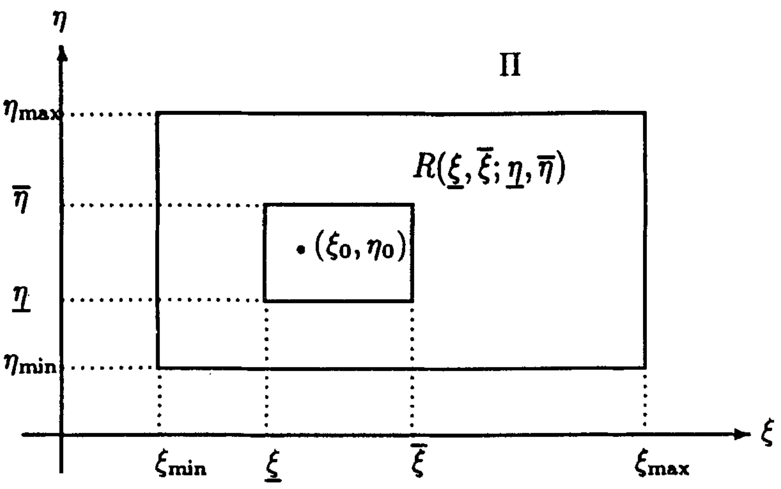

Fig. 1.

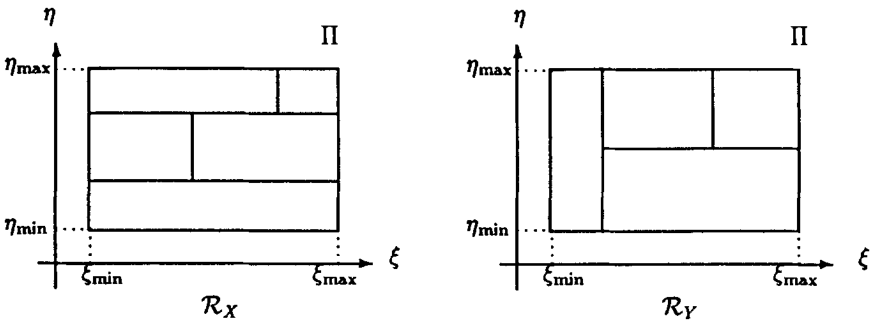

Fig. 2. Partition of the rectangle Π.

Let us now describe the outline of the procedure for searching the collection $\mathcal{R}_X$ of subrectangles.

```
Procedure Rectangle($P_X$)
  Input: A parametric linear programming problem $P_X(\xi,\eta)$
      $\xi_{\min}, \xi_{\max}, \eta_{\min}, \eta_{\max}$
  Output: The collection $\mathcal{R}_X$ of subrectangles
Step 1 (Initialization)
  solve a linear programming problem $P_X(\xi_{\min},\eta_{\min})$;
  let $B$ be an optimal basis of $P_X(\xi_{\min},\eta_{\min})$;
  set $\xi := \xi_{\min}$; $\hat{\xi} := \xi_{\max}$; $\eta := \eta_{\min}$; $\mathcal{P} := \emptyset$; $\mathcal{R}_X := \emptyset$;
```

```
Step 2
iterate:
  while $\xi < \hat{\xi}$
    begin
    calculate a pair of intervals $[\xi, \bar{\xi}]$, $[\eta, \bar{\eta}]$, and a new basis $B$
        such that $B$ is an optimal basis of the problem $P_X(\xi, \eta)$;
        $\mathcal{R}_X := \mathcal{R}_X \cup R(\underline{\xi}; \bar{\xi}; \underline{\eta}, \bar{\eta})$;
    if $\bar{\eta} < \eta_{\max}$ then
        $\mathcal{P} := \mathcal{P} \cup \{(P_X(\xi, \bar{\eta}), B, \bar{\xi})\}$;
      (Comment: $B$ is also an optimal basis of $P_X(\xi, \bar{\eta})$)
    $\xi := \bar{\xi}$;
    end
  if $\mathcal{P} \neq \emptyset$ then
    begin
        $(P_X(\xi, \eta), B, \hat{\xi}) := \min_{\eta} \mathcal{P}$;
        $\mathcal{P} := \mathcal{P} \setminus \{(P_X(\xi, \eta), B, \hat{\xi})\}$;
    goto iterate;
    end
  else (Comment: $\mathcal{P} = \emptyset$)
    terminate □
```

2.3. CALCULATIONS OF THE GLOBAL MINIMUM

Let us next describe how to calculate the global minimum of $h(\xi, \eta)$ over $(\xi, \eta) \in \Pi$.

Let H_X and V_X be the set of horizontal and perpendicular line segments associated with the partition $\mathcal{R}_X$. Also let H_Y and V_Y be the set of horizontal and vertical lines associated with $\mathcal{R}_Y$. Further, let $\mathcal{R}_{XY}$ be the partition of Π generated by the union $H_X \cup H_Y$ and $V_X \cup V_Y$. (See Figure 3.)

THEOREM 2.1. *The global minimum of $h(\xi, \eta)$ over $(\xi, \eta) \in \Pi$ is attained at either one of the vertices of the rectangle contained in $\mathcal{R}_{XY}$.*

Proof. $h_X(\xi, \eta)$ and $h_Y(\xi, \eta)$ are both bilinear function of (ξ, η). Theorem 1.2 asserts that the global minimum of $h(\xi, \eta)$ over R is attained at either one of its vertices. □

Let us note that $\mathcal{R}_{XY}$ may consist of a huge number of subrectangles even if the configurations of $\mathcal{R}_X$ and $\mathcal{R}_Y$ are relatively simple. However, we can ignore most of the vertices of $\mathcal{R}_{XY}$ by exploiting the special structure of the problem.

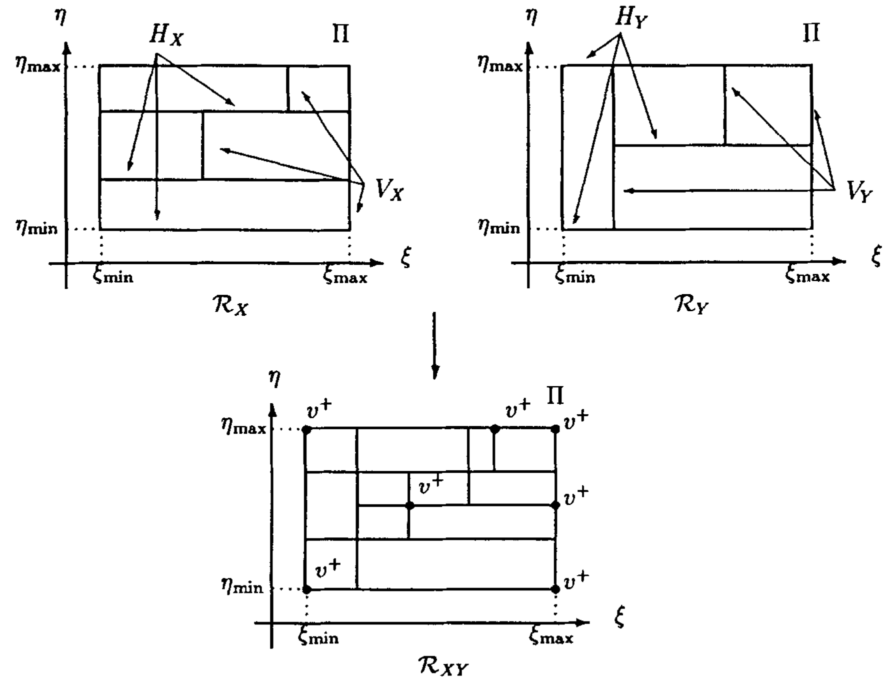

Fig. 3.

DEFINITION. A function $g(\xi, \eta)$ is a convex-concave (concave-convex) function if

(i) $g(\xi, \eta)$ is a convex (concave) function of ξ for fixed value of η.
(ii) $g(\xi, \eta)$ is a concave (convex) function of η for fixed value of ξ.

THEOREM 2.2. *$h_X(\xi, \eta)$ is a convex-concave function on Π. Also $h_Y(\xi, \eta)$ is a concave-convex function on Π.*

Proof. Follows from the standard results of linear programming. See, e.g. [1] □

THEOREM 2.3. *Let V^+ be the set of vertices which are intersections of linear segments of V_X and H_Y. Then only the vetices $v^+ \in V^+$ can be a local minimum of $h(\xi, \eta)$ over $(\xi, \eta) \in \Pi$.*

Proof. Since $h_X(\xi, \eta)$ is a concave function of η, any vertex lying on the line segment H_X, not on the H_Y, cannot be a local minimum of $h(\xi, \eta)$. Neither can any vertex lying on the line segment V_Y be a local minimum. □

For example, three vertices out of the four newly generated vertices of Figure 3 can be ignored throughout our effort to locate the global minimum of $h(\xi, \eta)$.

From Theorem 2.3, searching only the vertices $v^+ \in V^+$, suffices to obtain a global minimum point (ξ^*, η^*). By executing the procedure *Rectangle* twice, we can obtain the set of the line segments V_X, H_Y. Also the intersection of V_X and H_Y can be calculated efficiently by using plane sweep method [11].

2.4. COMPUTATIONAL EXPERIMENTS

Let us show the results of computational experiments. We solved the following rank two bilinear programming problems:

$$\left|\begin{array}{ll} \text{minimize} & c_0^t x + d_0^t y + c_1^t x \cdot d_1^t y + c_2^t x \cdot d_2^t y \\ \text{subject to} & A_1 x \leq b_1\,, \quad x \geq 0\,, \\ & A_2 y \leq b_2\,, \quad y \geq 0\,, \end{array}\right.$$

where $x, y \in R^n$, $A_1, A_2 \in R^{m\times n}$, $c_1, c_2, d_1, d_2, c_0, d_0 \in R^n$, $b_1, b_2 \in R^m$.

The program was coded in C language and tested on a SUN4/280S computer. All elements of A_1, A_2, b_1, b_2 are randomly generated, whose ranges are $[0, 100]$. Also c_1, c_2, d_1, d_2, c_0, d_0 are randomly generated, whose ranges are $[-50, 50]$. Ten problems were solved for each size of m, n. In Table II.1, Stage 1 corresponds to the total time of Step1 of the procedure *Rectangle*, i.e. the time required to solve two linear programming problems $P_X(\xi, \eta)$ and $P_Y(\xi, \eta)$. Stage 2 corresponds to Step 2 of the procedure *Rectangle*, i.e. the time spent to partition Π into its subrectangles $\mathcal{R}_X$ and $\mathcal{R}_Y$, while Stage 3 stands for the time required to calculate the intersections of V_X and H_Y.

We see from this table that the total amount of computation time is about three times more than that of Stage 1 for this class of problems, regardless of its size. This is a remarkable improvement over the results reported in [9], where the amount of computation grew exponentially as a function of n and m. Considering the magnitude of standard deviation, we may as well claim that rank two bilinear problems can now be solved in not more than five times as much computation time as that needed to solve the associated linear programs.

3. Three Parameter Simplex Algorithm for Rank Three Bilinear Problems

3.1. PARAMETRIC REPRESENTATION OF THE PROBLEM

In this section, we will consider a special class of rank three bilinear programming problem in which $d_0 = 0$, i.e.,

$$\left|\begin{array}{ll} \text{minimize} & G_3(x, y) = c_0^t x + c_1^t x \cdot d_1^t y + c_2^t x \cdot d_2^t y + c_3^t x \cdot d_3^t y \\ \text{subject to} & x \in X\,, \quad y \in Y\,, \end{array}\right. \tag{3.1}$$

where, as before, we assume that X and Y are nonempty and bounded polyhedral

Table II.1. General rank two bilinear programming problem

m	100	100	100	150	150	150	200	200	200
n	80	100	120	120	150	180	180	200	220
Average CPU time (standard deviation)									
Stage 1	21.32	24.70	30.24	78.54	91.51	100.43	197.04	205.67	195.30
	(2.46)	(2.72)	(4.81)	(6.68)	(13.32)	(10.86)	(24.01)	(26.89)	(89.87)
Stage 2	24.48	39.25	62.02	117.70	172.94	210.52	371.27	334.60	419.16
	(9.08)	(12.27)	(25.37)	(52.60)	(88.85)	(41.07)	(132.46)	(160.60)	(243.04)
Stage 3	0.09	0.14	0.20	0.28	0.35	0.36	0.55	0.43	0.54
	(0.03)	(0.04)	(0.07)	(0.12)	(0.20)	(0.11)	(0.24)	(0.22)	(0.34)
Total	45.89	64.09	92.46	196.52	264.80	311.32	568.87	540.71	738.00
	(10.01)	(14.59)	(29.47)	(56.43)	(101.30)	(49.65)	(150.71)	(184.38)	(186.81)
Average Number of Segments (standard deviation)									
V_X	184.70	335.00	473.44	602.30	794.90	757.80	1126.50	873.50	987.67
	(64.76)	(93.89)	(162.96)	(236.48)	(415.67)	(285.22)	(604.71)	(333.49)	(643.21)
H_Y	356.50	372.00	459.78	611.90	654.80	761.10	883.50	822.00	932.00
	(203.61)	(189.10)	(376.74)	(373.33)	(314.24)	(130.48)	(366.55)	(516.55)	(646.48)
Average Number of intersections of V_Y and H_X (standard deviation)									
	348.10	503.60	638.56	853.10	1013.10	1063.90	1446.60	1367.60	1462.92
	(133.99)	(131.90)	(260.97)	(354.39)	(430.27)	(206.20)	(425.83)	(659.83)	(772.60)

convex sets. By definition, $\{c_1, c_2, c_3\}$ and $\{d_1, d_2, d_3\}$ are linearly independent sets of vectors. Also, let us assume that c_0 is linearly independent of $\{c_1, c_2, c_3\}$.

Let us introduce three auxiliary variables

$$\begin{cases} \xi = c_1^t x \,, \\ \eta = d_2^t y \,, \\ \zeta = c_3^t x \,, \end{cases} \tag{3.2}$$

and consider parametric representation of (3.1):

$$\left| \begin{array}{ll} \text{minimize} & g_3(x, y; \xi, \eta, \zeta) = c_0^t x + \xi d_1^t y + \eta c_2^t x + \zeta d_3^t y \\ \text{subject to} & A_1 x = b_1 \,, \quad A_2 y = b_2 \,, \\ & c_1^t x = \xi \,, \quad d_2^t y = \eta \,, \\ & c_3^t x = \zeta \,, \quad x \geq 0 \,, \quad y \geq 0 \,, \\ & (\xi, \zeta) \in \Pi_2 \,, \quad \eta_{\min} \leq \eta \leq \eta_{\max} \,, \end{array} \right. \tag{3.3}$$

where

$$\begin{cases} \eta_{\min} = \min\{d_2^t y \mid y \in Y\} \,, \\ \eta_{\max} = \max\{d_2^t y \mid y \in Y\} \,, \end{cases} \tag{3.4}$$

and Π_2 is a collection of (ξ, ζ) such that

$$X(\xi, \zeta) = \{x \mid A_1 x = b_1 \,, \quad c_1^t x = \xi \,, \quad c_3^t x = \zeta \,, \quad x \geq 0\} \tag{3.5}$$

is nonempty.

Let

$$\Pi_3 = \{(\xi, \eta, \zeta) \mid (\xi, \zeta) \in \Pi_2 \,, \quad \eta_{\min} \leq \eta \leq \eta_{\max}\} \,. \tag{3.6}$$

3.2. PARTITION OF THE THREE DIMENSIONAL PARAMETER SPACE

Let us define a subproblem:

$$P(\xi, \eta, \zeta) \left| \begin{array}{ll} \text{minimize} & g_3(x, y; \xi, \eta, \zeta) = c_0^t x + \xi d_1^t y + \eta c_2^t x + \zeta d_3^t y \\ \text{subject to} & A_1 x = b_1 \,, \quad A_2 y = b_2 \,, \\ & c_1^t x = \xi \,, \quad d_2^t y = \eta \,, \\ & c_3^t x = \zeta \,, \\ & x \geq 0 \,, \quad y \geq 0 \end{array} \right. \tag{3.7}$$

for each $(\xi, \eta, \zeta) \in \Pi_3$ and let $h(\xi, \eta, \zeta)$ be the minimal value of the objective function of $P(\xi, \eta, \zeta)$. As before, this problem can be decomposed into two subproblems:

$$P_X(\xi, \eta, \zeta) \left| \begin{array}{ll} \text{minimize} & c_0^t x + \eta c_2^t x \\ \text{subject to} & A_1 x = b_1 \,, \quad x \geq 0 \,, \\ & c_1^t x = \xi \,, \\ & c_3^t x = \zeta \,, \end{array} \right. \tag{3.8}$$

$$P_Y(\xi,\eta,\zeta)\left|\begin{array}{ll}\text{minimize} & \xi d_1^t y + \zeta d_3^t y \\ \text{subject to} & A_2 y = b_2\,, \quad y \geqslant 0\,, \\ & d_2^t y = \eta\,.\end{array}\right. \tag{3.9}$$

Let $h_X(\xi,\eta,\zeta)$ and $h_Y(\xi,\eta,\zeta)$ be the optimal value of $P_X(\xi,\eta,\zeta)$ and $P_Y(\xi,\eta,\zeta)$, respectively. Now let us consider $P_Y(\xi_0,\eta_0,\zeta_0)$ where $(\xi_0,\eta_0,\zeta_0) \in \Pi_3$. Note that $P_Y(\xi_0,\eta_0,\zeta_0)$ is a linear program.

Let B_0 be the optimal basis of the linear program (3.9), and N_0 be the associated nonbasic matrix. Then

$$B_0^{-1}\begin{pmatrix} b_2 \\ \eta_0 \end{pmatrix} \geqslant 0\,,$$

and

$$\xi_0 \bar{d}_{1N_0} + \zeta_0 \bar{d}_{3N_0} \geqslant 0\,,$$

where

$$\bar{d}_{1N_0}^t = d_{1N_0}^t - d_{1B_0}^t B_0^{-1} N_0\,,$$

$$\bar{d}_{3N_0}^t = d_{3N_0}^t - d_{3B_0}^t B_0^{-1} N_0\,.$$

Also, B_0 is an optimal basis of $P_Y(\xi,\eta,\zeta)$ for all (ξ,η,ξ) such that

$$B_0^{-1}\begin{pmatrix} b_2 \\ \eta \end{pmatrix} \geqslant 0\,,$$

and

$$\xi \bar{d}_{1N_0} + \zeta \bar{d}_{3N_0} \geqslant 0\,.$$

Thus we obtain a region[1]

$$S(\underline{\eta},\bar{\eta};\underline{a},\bar{a}) \equiv \{(\xi,\eta,\zeta) \mid \underline{\eta} \leqslant \eta \leqslant \bar{\eta},\ \underline{a}\xi \leqslant \zeta \leqslant \bar{a}\xi\} \cap \Pi_3$$

in which B_0 remains an optimal basis of $P_Y(\xi,\eta,\zeta)$.

Let $S_0 = S(\underline{\eta},\bar{\eta};\underline{a},\bar{a})$. When (ξ,η,ζ) reaches the boundary $(\xi,\eta,\underline{a}\xi)$, we obtain an adjacent optimal basis B_1 by a primal simplex iteration. Let S_1 be the associated subregion of Π_3 in which B_1 remains optimal. In this way, we will obtain a clockwise partition $S_0, S_1, \ldots, S_{k-1}$ of Π_3 by cones associated with a sequence of bases $B_0, B_1, \ldots, B_{k-1}, B_k = B_0$, barring degeneracy appropriately. (See Figure 4 and [10].) Furthermore, when (ξ,η,ζ) reaches the boundary $(\xi,\bar{\eta},\zeta)$, we will obtain alternative basis by a dual simplex iteration and the region in which this basis remains optimal.

Thus we obtain a partition $\mathscr{S}$ of Π_3 into a finite number of subregions in which certain base B of the matrix $\binom{A_2}{d_2}$ remains optimal. Each subregion has the shape like a slice of cheese-cake. (Figure 5).

For simplicity, we split Π_3 into four subsets:

$$\Pi_3^{++} = \{(\xi,\eta,\zeta) \mid \xi \geqslant 0\,,\quad \eta \geqslant 0\,,\quad (\xi,\zeta) \in \Pi_2\,,\quad \eta_{\min} \leqslant \eta \leqslant \eta_{\max}\}\,,$$

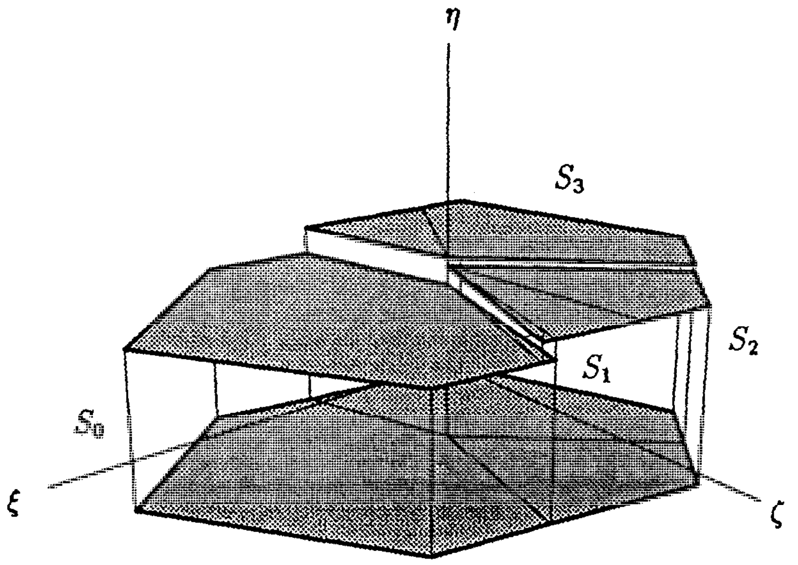

Fig. 4. An Example of a Clockwise Partition of Π_3

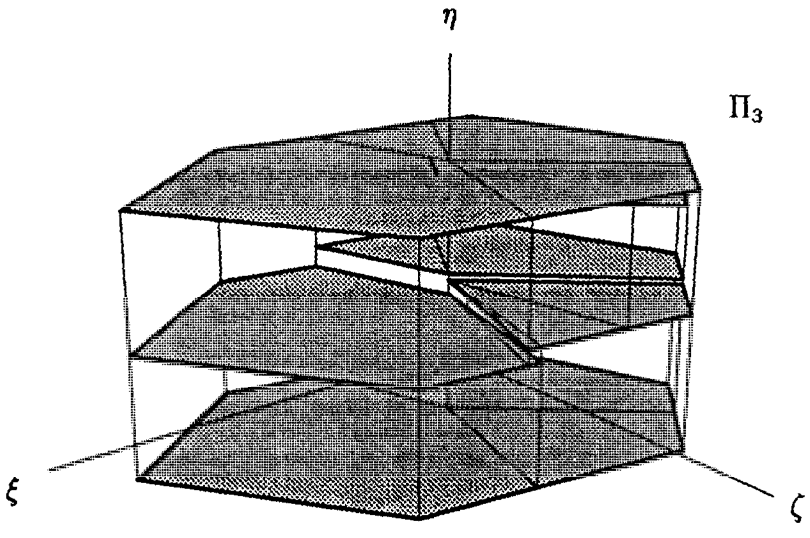

Fig. 5. An Example of a Complete Partition of Π_3.

$$\Pi_3^{+-} = \{(\xi, \eta, \zeta) \mid \xi \geq 0, \quad \eta \leq 0, \quad (\xi, \zeta) \in \Pi_2, \quad \eta_{\min} \leq \eta \leq \eta_{\max}\},$$
$$\Pi_3^{-+} = \{(\xi, \eta, \zeta) \mid \xi \leq 0, \quad \eta \geq 0, \quad (\xi, \zeta) \in \Pi_2, \quad \eta_{\min} \leq \eta \leq \eta_{\max}\},$$
$$\Pi_3^{--} = \{(\xi, \eta, \zeta) \mid \xi \leq 0, \quad \eta \leq 0, \quad (\xi, \zeta) \in \Pi_2, \quad \eta_{\min} \leq \eta \leq \eta_{\max}\},$$

and partition each subset separately.

Assume $\Pi_3^{++} \neq \emptyset$ and let us consider partitioning Π_3^{++} into subcollection $\mathcal{S}^{++}$ of $\mathcal{S}$. In this case, we can reduce $P_Y(\xi, \eta, \zeta)$ into the two parameter problem by dividing the objective function of $P_Y(\xi, \eta, \zeta)$ by ξ.

$$P_Y^{++}(\sigma, \eta)\left|\begin{array}{ll} \text{minimize} & d_1^t y + \sigma d_3^t y \\ \text{subject to} & A_2 y = b_2\,, \quad y \geq 0\,, \\ & d_2^t y = \eta\,, \end{array}\right. \tag{3.10}$$

where $\sigma \in [0, \infty]$, $\eta \in [\eta_{\min}, \eta_{\max}]$. Using procedure *Rectangle*(P_Y^{++}) presented in Section 2.2, two parameter region

$$\{(\sigma, \eta) \mid 0 \leq \sigma \leq \infty, \eta_{\min} \leq \eta \leq \eta_{\max}\}$$

can be partitioned into the collection of subrectangles. As before, each subregion can be expressed as follows:

$$\{(\sigma, \eta) \mid \underline{\sigma} \leq \sigma \leq \bar{\sigma}, \underline{\eta} \leq \eta \leq \bar{\eta}\}\,,$$

which corresponds to

$$S(\underline{\eta}, \bar{\eta}; \underline{\sigma}, \bar{\sigma}) = \{(\xi, \eta, \zeta) \mid \underline{\eta} \leq \eta \leq \bar{\eta}, \underline{\sigma}\xi \leq \zeta \leq \bar{\sigma}\xi\} \cap \Pi_3\,.$$

The other subsets, Π_3^{+-}, Π_3^{-+}, or Π_3^{--} can be partitioned analogously.

3.3. ALGORITHM TO OBTAIN A GLOBAL MINIMUM

Let S be a member of a partition $\mathcal{S}$ and let B be the associated basis matrix of $\binom{A_2}{d_2}$. An optimal solution $y^*(\eta)$ of $P_Y(\xi, \eta, \zeta)$ for $(\xi, \eta, \zeta) \in S$ is given by

$$y_B^*(\eta) = B^{-1}\begin{pmatrix} b_2 \\ \eta \end{pmatrix}, \quad y_N^*(\eta) = 0\,,$$

and we obtain the following expression

$$h_Y(\xi, \eta, \zeta) = (\xi d_{1B} + \zeta d_{3B})^t B^{-1}\begin{pmatrix} b_2 \\ \eta \end{pmatrix}, \quad (\xi, \eta, \zeta) \in S\,.$$

THEOREM 3.1. *Global minimum of $h(\xi, \eta, \zeta)$ over S is attained at either $\eta = \underline{\eta}$ or $\eta = \bar{\eta}$.*

Proof. $h_Y(\xi, \eta, \zeta)$ is a linear function of η on S. Thus the minimum of $h_Y(\xi, \eta, \zeta)$ over S is attained at $\eta = \underline{\eta}$ or $\eta = \bar{\eta}$. Also, since η appears linearly in $P_X(\xi, \eta, \zeta)$ only in its objective function, $h_X(\xi, \eta, \zeta)$ is a piecewise linear concave function of η. Therefore, the global minimum of $h(\xi, \eta, \zeta)$ over S is attained at either one of its extreme points, i.e., $\eta = \underline{\eta}$ or $\eta = \bar{\eta}$. □

Thus to obtain a global minimum of $h(\xi, \eta, \zeta)$ over Π_3, we need only consider the finite set of points

$$Y_{ext} \equiv \{y^*(\underline{\eta}) \mid S(\underline{\eta}, \bar{\eta}; \underline{a}, \bar{a}) \in \mathcal{S}\} \cup \{y^*(\bar{\eta}) \mid S(\underline{\eta}, \bar{\eta}; \underline{a}, \bar{a}) \in \mathcal{S}\}$$

As a result, (3.1) can be reduced to a finite sequence linear programming problems:

$$\left| \begin{array}{ll} \text{minimize} & (d_1^t y \cdot c_1 + d_2^t y \cdot c_2 + d_3^t y \cdot c_3 + c_0)^t x \\ \text{subject to} & x \in X , \end{array} \right. \tag{3.11}$$

where $y \in Y_{ext}$.

We solve these linear programming problems in Step 2 of procedure *Rectangle*(P_Y) whenever a new subregion S is generated. Since these linear programming problems differ only in their cost vectors, the preceding optimal basic solution can be used as the initial feasible solution.

3.4. COMPUTATIONAL EXPERIMENTS

We will show the results of the numerical experiments. We solved following rank three bilinear programming problems:

$$\left| \begin{array}{ll} \text{minimize} & c_0^t x + c_1^t x \cdot d_1^t y + c_2^t x \cdot d_2^t y + c_3^t x \cdot d_3^t y \\ \text{subject to} & A_1 x \leq b_1 , \quad x \geq 0 , \\ & A_2 y \leq b_2 , \quad y \geq 0 , \end{array} \right.$$

where $x, y \in R^n$, $A_1, A_2 \in R^{m \times n}$, $c_1, c_2, c_3, d_1, d_2, d_3, c_0 \in R^n$, $b_1, b_2 \in R^m$.

The program was again coded in C language and tested on a SUN4/280S computer. All elements of A_1, A_2, b_1, b_2 are randomly generated, whose ranges are $[0, 100]$. Also $c_1, c_2, c_3, d_1, d_2, d_3, c_0$ are randomly generated, whose ranges are $[-50, 50]$. Ten problems were solved for each size of m, n. Table III.1 shows the statistics of the experiments. In this table, Stage 1 corresponds to Step 1 of procedure *Rectangle*, i.e., the procedure to solve initial linear programming problem P_Y. Stage 2 corresponds to Step 2, and including the process of solving linear programming problems (3.11).

Table III.1 shows that the computation time grows exponentially as a function of the size of the problem. It appears, however, that the number of subregions grows much slower for this class of problems.

Moreover, note that the ratio of (a) and (b) is almost constant. This is due to the fact that the sequence of linear programming problems (3.11) are not much different from the one solved in the preceding iteration.

4. Conclusion

We showed in this paper that solving rank two bilinear programs are not more difficult than solving the associated linear programming problem.

On the other hand, solving a special class of rank three bilinear programming problems (problems in which $c_0 = 0$ or $d_0 = 0$) requires 30 times more computation times than that needed to solve the associated linear program. Though not as efficient as the algorithm for rank two problems, we believe that this algorithm is

Table III.1. Rank three bilinear programming problem

m	50	50	50	100	100	100	120	120	120
n	40	50	60	80	100	120	100	120	140
Average CPU time (standard deviation)									
Stage 1	1.40	2.02	1.98	10.75	13.66	15.48	17.94	23.08	26.17
	(0.17)	(0.37)	(0.47)	(1.31)	(1.68)	(2.64)	(1.99)	(3.03)	(3.75)
Stage 2	20.66	43.42	46.43	183.65	412.86	410.22	486.11	689.39	874.92
	(12.70)	(24.00)	(24.28)	(70.37)	(182.63)	(189.22)	(386.33)	(448.39)	(380.20)
Total	22.06	45.43	48.41	194.40	426.51	425.70	504.06	712.47	901.09
	(12.84)	(24.27)	(24.69)	(71.08)	(182.93)	(191.14)	(387.68)	(450.47)	(382.77)
(a) Average number of pivots to solve a sequence of LPs (3.11) (standard deviation)									
	1505.90	2417.30	2468.30	39.42.10	7753.44	6177.40	7809.20	9236.00	9732.70
	(1030.93)	(1340.89)	(1408.61)	(1743.70)	(3871.35)	(3025.85)	(6499.76)	(6414.07)	(4423.15)
(b) Average Number of Subregions (standard deviation)									
	358.30	709.90	529.90	769.00	1188.56	1128.60	1060.90	1304.70	1513.80
	(207.35)	(426.75)	(253.27)	(278.15)	(414.85)	(549.17)	(767.93)	(713.31)	(637.17)
(a)/(b) (standard deviation)									
	4.25	3.81	4.45	5.28	6.47	5.64	6.98	6.77	6.67
	(1.25)	(1.23)	(1.39)	(1.65)	(2.38)	(1.96)	(1.26)	(2.27)	(1.76)

reasonably efficient considering the intrinsic difficulty of this class of problems. We are now planning a more extensive numerical experiments using the problems with different data structure, whose results will be reported elsewhere.

An exact parametric algorithm for general rank three problems (the problem in which $c_0 \neq 0$, $d_0 \neq 0$) is expected to requires even more computation time. Thus to solve higher rank problems, we need to develop some approximation scheme, which will be discussed in detail in a subsequent paper.

Note

[1]We assume that $S(\underline{\eta}, \bar{\eta}; \underline{a}, \bar{a}) \subseteq \{(\xi, \eta, \zeta) \mid \xi \geqslant 0,\ (\xi, \zeta) \in \Pi_2,\ \eta_{\min} \leqslant \eta \leqslant \eta_{\max}\}$.

References

1. Chvátal, V. (1983), *Linear Programming*, W. H. Freeman and Company.
2. Falk, J. E. (1973), A Linear Max-Min Problem, *Mathematical Programming* **5**, 169–188.
3. Falk, J. E. and Hoffman, K. R. (1976), A Successive Underestimation Method for Concave Minimization Problems, *Math. Oper. Res.* **1**, 251–259.
4. Gallo, G. and Ülkücü, A. (1977), Bilinear Programming: An Exact Algorithm, *Mathematical Programming* **12**, 173–194.
5. Horst, R. and Tiuy, H. (1990), *Global Optimization*, Springer-Verlag.
6. Konno, H. (1971), Bilinear Programming, Part 1 and 2, Technical Report No. 71-9, 71-10, Dept. of OR, Stanford University.
7. Konno, H. (1976), A Cutting Plane Algorithm for Solving Bilinear Programs, *Mathematical Programming* **11**, 14–27.
8. Konno, H. (1976), Maximization of a Convex Quadratic Function under Linear Constraints, *Mathematical Programming* **11**, 117–127.
9. Konno, H. and Yajima, Y. (1990), Solving Rank Two Bilinear Programs by Parametric Simplex Algorithms, IHSS Report 90-17, Institute of Human and Social Sciences, Tokyo Institute of Technology.
10. Konno, H., Yajima, Y., and Matsui, T. (1990), Parametric Simplex Algorithm for Solving a Special Class of Nonconvex Minimization Problems, IHSS Report 90-16, Institute of Human and Social Sciences, Tokyo Institute of Technology. (See also *J. of Global Optimization* **1**, 65–81).
11. Lee, D. T. and Preparata, F. P. (1984), Computational Geometry – a Survey, *IEEE Transactions on Computers* **c-33**, 1072–1101.
12. Majthay, A. and Whinston, A. B. (1974), Quasi-Concave Minimization Subject to Linear Constraints, *Discrete Math.* **9**, 35–59.
13. Sherali, H. and Shetty, C. M. (1980), A Finitely Convergent Algorithm for Bilinear Programming Problems Using Polar Cuts and Disjunctive Face Cuts, *Mathematical Programming* **19**, 14–31.
14. Thaoi, N. V. and Tuy, H. (1980), Convergent Algorithms for Minimizing Concave Function, *Math. Oper. Res.* **5**, 556–566.
15. Tuy, H. (1964), Concave Programming under Linear Constraints, *Soviet Mathematics* **5**, 1437–1440.
16. Tuy, H., Thieu, T. V., and Thaoi, N. Q. (1985), A Conical Algorithm for Globally Minimizing a Concave Function over a Closed Convex Set, *Math. Oper. Res.* **10**, 489–514.
17. Vaish, H. and Shetty, C. M. (1976), The Bilinear Programming Problem, *NRLQ* **23**, 303–309.
18. Vaish, H. and Shetty, C. M. (1977), A Cutting Plane Algorithm for Bilinear Programming Problem, *NRLQ* **24**, 83–94.