DEPARTMENT OF OPERATIONS RESEARCH
COLLEGE OF ENGINEERING
CORNELL UNIVERSITY
ITHACA, NEW YORK

TECHNICAL REPORT NO. 195

August 1973

COMPLEMENTARITY ALGORITHMS WITHOUT RAYS

by

Michael J. Todd

ABSTRACT

The linear complementarity problem is to find w,z with $w = Mz + q$; $w,z \geq 0$; and $w^T z = 0$. Lemke's algorithm to solve this problem deals with the primary ray, secondary rays and complementary solutions, corresponding to the starting solution, unsuccessful completion and successful completion. We show that a simple bounding constraint turns these into the starting (undesirable) complementary solution, other undesirable complementary solutions, and desirable complementary solutions respectively. The advantage lies in an algorithm, extending Lemke's, which uses any undesirable complementary solutions found (corresponding to failure of Lemke's algorithm) as starting points of new almost complementary paths. We also reformulate the bimatrix game problem to permit solution using pivoting in bounded linear systems; in this case, there is only one undesirable complementary solution, which is the starting solution.

1. Introduction

The linear complementarity problem (LCP) is to find w,z satisfying

$$w = Mz + q, \quad w \geq 0, \quad z \geq 0, \quad \text{and} \tag{1}$$

$$w^T z = 0. \quad (M \text{ is } n\times n;\ w, z \text{ and } q \ n\times 1.) \tag{2}$$

The LCP has been widely studied; see, for example, [2-5, 7, 9-12].

Lemke's algorithm (more precisely, his scheme I: see [10]) to solve this problem is based on adjoining an extra z variable, to get

$$w = ez_0 + Mz + q, \quad w \geq 0, \quad z_0 \geq 0, \quad z \geq 0. \tag{3}$$

(We use e for a vector of 1's of appropriate dimension.) This device enables a starting solution (adjacent to a primary ray) to be found, from which an almost complementary path may find a true complementary solution. We presume the reader to be familiar with this algorithm.

Our device uses not only an extra z-variable, but also an extra constraint and associated w-variable. This extra constraint is very similar (the coefficient of 1 rather than 0 for z_0 being the only difference) to one mentioned briefly in [9].

Consider solutions $\bar{w},\bar{z}$ to

$$\bar{w} = \bar{M}\bar{z} + \bar{q}, \quad \bar{w} \geq 0, \quad \bar{z} \geq 0; \tag{4}$$

$$\bar{w}^T\bar{z} = 0; \tag{5}$$

where $\bar{w} = \begin{pmatrix} w_0 \\ w \end{pmatrix}$, $\bar{z} = \begin{pmatrix} z_0 \\ z \end{pmatrix}$ and $\bar{q} = \begin{pmatrix} k \\ q \end{pmatrix}$ are $(n+1)\times 1$, and $\bar{M} = \begin{pmatrix} -1 & -e^T \\ e & M \end{pmatrix}$ is $(n+1)\times(n+1)$.

The scalar k is chosen so that all basic solutions to (3) satisfy $z_0 + e^T z < k$. Then any solution to (1),(2) gives a solution to (4),(5) by setting $z_0 = 0$, $w_0 = k - e^T z$. Conversely, solutions to (4),(5) may have $w_0 > 0$ and hence $z_0 = 0$. These give solutions to (1),(2). Otherwise $z_0 > 0$ and $w_0 = 0$, and we do not have a solution to (1),(2).

Let us note briefly that a similar bounding constraint can be used in the generalization of the LCP due to Scarf [13], and that due to Cottle and Dantzig [3], with similar advantages.

We assume throughout that (1) and (4) are non-degenerate systems, i.e. that no solution has more than n (for (1)) or $n + 1$ (for (4)) zero-valued variables. This can be ensured by perturbation of q or standard lexicographic techniques.

A solution to (4) is called basic if it has exactly $n + 1$ zero-valued variables. It is complementary if it also satisfies (5), and i-almost complementary (i-a.c.) if it satisfies $\bar{w}^T\bar{z} = w_i z_i$, for $i = 0,1,\ldots n$. A complementary solution is desirable (a d.c. solution) if $z_0 = 0$, and undesirable (a u.c. solution) if $w_0 = 0$.

In §2, we give two algorithms which may find a solution to the LCP, and which extend that of Lemke [10]. These algorithms use any u.c. solutions found as new starting solutions, if necessary, of new almost complementary paths. We give examples to show the superiority of our algorithms, and the possibility of its failure even when a d.c. solution exists.

In §3, we discuss the application of these algorithms to two classes of matrices mentioned in [11] and [12]. For one class, we show Lemke's algorithm

always succeeds; for the other, both our algorithms succeed while Lemke's fails.

In §4, we briefly show how these ideas can be used to reformulate the problem of finding Nash equilibrium points of bimatrix games, and the resulting algorithm which again avoids rays. In this case, the only advantage is in simplicity; no computational short-cuts result.

Finally, in §5, we discuss two general techniques for dealing with incomplete pivoting structures. We indicate how the algorithms of this paper provide an argument for one of these techniques, that of artificial completion of the structures.

2. The Algorithms

First note that transforming (1),(2) into (4),(5) has given us a readily available u.c. solution, called the initial solution:

(6) $$w_0 = 0, \quad w = ke + q, \quad z_0 = k, \quad z = 0.$$

The basic step of both algorithms is similar to that in all complementary pivot algorithms to solve the LCP. To explain it simply, define, for $i = 0,1,\ldots n$, the graph G_i to have as nodes all almost-complementary solutions. Two almost-complementary solutions $(\bar{w}^1,\bar{z}^1)$ and $(\bar{w}^2,\bar{z}^2)$ are adjacent in G_i if:

(7) both are i-a.c.;

(8) $\begin{pmatrix}\bar{w}^1\\ \bar{z}^1\end{pmatrix}$ and $\begin{pmatrix}\bar{w}^2\\ \bar{z}^2\end{pmatrix}$ have zeroes in the same coordinates in all except two cases.

Exactly as in Lemke's arguments we can show that every node of G_i has degree 0, 1, or 2, according as the corresponding solution is not i-a.c., complementary, or i-a.c. but not complementary. Note here that the extra constraint in (4) has rendered it a bounded system. Thus all pivots are "blocked" and can be completed; there are no rays.

The graph G is just the union of the G_i; i.e., its nodes are the nodes of G_0 (say), and any edge of any G_i is an edge of G. No edge can be in both G_i and G_j for $i \neq j$. Thus each edge of G can be labeled as the graph it occurs in.

In the graph G, each node has degree 0, 2 or n +1, according as it is not almost-complementary, almost-complementary but not complementary, or complementary.

By "search G_i from $(\bar{w},\bar{z})$ to find $(\bar{w}^{-1},\bar{z}^{-1})$" we mean trace the unique path of i-a.c. solutions in G_i from the complementary solution $(\bar{w},\bar{z})$ until the complementary solution $(\bar{w}^{-1},\bar{z}^{-1})$ is found.

We now state our first algorithm:

Algorithm 1.

Step 1. Search G_0 from the initial solution given by (6) to find $(\bar{w}^{-1},\bar{z}^{-1})$. If $(\bar{w}^{-1},\bar{z}^{-1})$ is desirable, STOP. If not, it is a u.c. solution. Label this and the initial solution 0.

Step 2. Pick any labeled solution $(\bar{w},\bar{z})$ which is not labeled $0,1,2,\ldots,n$. Let j be any integer with which it is not labeled. Search G_j from $(\bar{w},\bar{z})$ to find $(\bar{w}^{-1},\bar{z}^{-1})$. If desirable, STOP. If not, label $(\bar{w},\bar{z})$ and $(\bar{w}^{-1},\bar{z}^{-1})$ j. If all labeled solutions are labeled $0,1,\ldots,n$, STOP. Otherwise return to Step 2.

The algorithm fails if it stops because all u.c. solutions found are

labeled $0,1,\ldots n$. This means it has searched all of one connected component of G. We have:

Theorem 1.

a) Step 1 of algorithm 1 is equivalent to Lemke's algorithm, termination in a u.c. solution being equivalent to termination in a secondary ray.

b) Algorithm 1 will find a d.c. solution if one exists in the same connected component of G as the initial solution.

Proof. b) follows from the complete search of the connected component of G containing the initial solution.

For (a), note that the first edge traversed from the initial solution in G_0 takes one to Lemke's starting solution, with z_0 equal to minus the most negative q coordinate, and all but one of the w-variables basic. From then, searching G_0 is similar to Lemke's algorithm, until the artificial constraint $w_0 = k - z_0 - e^T z$ is encountered. However, our assumption on k shows that without this constraint, there would be no blocking variable and hence a secondary ray. Also, since w_0 becomes non-basic at this step, we have just reached a complementary solution to (4) (in fact, a u.c. solution).

Part (a) of Theorem 1 shows that algorithm 1 is at least as powerful as Lemke's; the following example shows that it is stronger.

Example 1. (from example 3 of [2]) Let

$$M = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & -1 \end{pmatrix} \qquad q = \begin{pmatrix} 1 \\ -1+\epsilon \\ 3 \\ 1 \end{pmatrix}$$

(The ϵ is for non-degeneracy)

Searching G_0 from the initial solution gives the u.c. solution: $z_0 = \frac{k-3}{2}$, $w_1 = \frac{k-1}{2}$, $w_2 = \frac{k-5}{2} + \varepsilon$, $z_3 = \frac{k+3}{2}$, $w_4 = \frac{k-1}{2}$, all others 0. In other words, Lemke's algorithm fails by generating a second ray. However, searching G_4 from the initial solution gives the u.c. solution: $z_0 = \frac{k-1}{2}$, $w_1 = k + 1$, $w_2 = k - 1 + \varepsilon$, $w_3 = k + 3$, $z_4 = \frac{k+1}{2}$, all others 0; and searching G_0 from this solution gives the d.c. solution: $w_0 = k - 1$, $w_1 = 2$, $w_2 = \varepsilon$, $w_3 = 4$, $z_4 = 1$, all others 0.

The following example shows that algorithm 1 may fail to find a d.c. solution if one exists.

Example 2. Let $M = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & -9 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & -9 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & \frac{1}{2} & 0 \\ 1 & 1 & 1 & 1 & 1 & \frac{3}{2} & 0 \\ -9 & -9 & -9 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -9 & 1 & 1 & 1 \end{bmatrix}$

and $q = (-90,-90,-99,-98,-98\frac{1}{2},-90,-90)^T$. Let $k = 100$ (this satisfies our assumption). Then we have the following complementary solution:

a) $z_0 = 100$, $w_1 = w_2 = 10$, $w_3 = 1$, $w_4 = 2$, $w_5 = 1\frac{1}{2}$, $w_6 = w_7 = 10$, all others 0. This is the initial u.c. solution.

b) $z_0 = 98$, $z_1 = 1$, $w_2 = 10$, $w_3 = 1$, $w_4 = 1\frac{1}{2}$, $w_5 = 2$, $z_6 = 1$, $w_7 = 10$, all others 0. This is an u.c. solution.

c) $w_0 = 1$, $w_1 = 4$, $w_2 = 1\frac{1}{2}$, $z_3 = z_4 = \frac{9}{10}$, $z_5 = 95\frac{19}{20}$, $z_6 = \frac{1}{2}$, $z_7 = \frac{3}{4}$, all others 0. This is a d.c. solution.

d) $w_0 = \frac{5}{6}$, $w_1 = 4\frac{1}{6}$, $z_2 = \frac{11}{12}$, $w_3 = \frac{1}{6}$, $z_4 = \frac{11}{12}$, $z_5 = 95\frac{11}{12}$, $z_6 = \frac{1}{2}$, $z_7 = \frac{11}{12}$, all others 0. This is a d.c. solution.

G has the following form (with intermediate nodes omitted):

Our algorithm finds only the u.c. solutions (a) and (b), whereas (c) and (d) are desirable but unreachable.

The algorithm we have proposed above must remember all u.c. solutions it has generated. This can of course be simplified by recording only whether $w_i = 0$ or not for $i = 1...n$; then if this u.c. solution is needed later, it can be retrieved by reinversion as in the inverse matrix method of linear programming (see [1]).

However, the simplified algorithm below, which searches only a portion of the connected component of G containing the initial solution, has no memory requirements.

Algorithm 2.

Step 1. Set $i = 0$ and $(\bar{w},\bar{z})$ equal to the initial solution.

Step 2. Search G_i from $(\bar{w},\bar{z})$ to find $(\bar{w}^1,\bar{z}^1)$. If $(\bar{w}^1,\bar{z}^1)$ is desirable, STOP.

Step 3. If $i = n$, go to Step 4. Otherwise, let $(\bar{w}^1,\bar{z}^1)$ replace $(\bar{w},\bar{z})$, increase i by 1 and go to Step 2.

Step 4. If $(\bar{w}^1,\bar{z}^1)$ has $z_1 = \ldots = z_n = 0$, STOP. Otherwise, let $(\bar{w}^1,\bar{z}^1)$ replace $(\bar{w},\bar{z})$, set $i = 0$ and go to Step 2.

The STOP in Step 4 indicates failure of algorithm 2. Because of Steps 1 and 2, Lemke's algorithm would certainly fail on such a problem. However, algorithm 1 could solve the problem by searching G more thoroughly.

Consideration of the graph G shows that no edge can be traversed twice in the same direction. However, it is possible that an edge (in fact, a whole i-a.c. path) can be traversed twice, in two different directions. This is not particularly harmful, as the path will diverge on the next reversion to Step 2.

Algorithm 2 will always search at least $n + 1$ a.c. paths before failure, and is recommended as a first trial to solution of any problem. If it fails, one can decide whether algorithm 1 should be tried.

Much research has been devoted to the range of successful applicability of Lemke's algorithm (see [2, 5, 11, 12]). Many important classes of matrices M have been found for which Lemke's algorithm will find a solution if one exists; more to the point, there are many applications of the LCP for which M is known a priori to lie within these classes. Generally it is simpler to apply the algorithm hopefully than attempt to verify whether a certain matrix lies in one of these classes. The obvious question arises as to what further classes of M can be processed by algorithms 1 and 2.

The structure of algorithm 2 makes an answer in this case very hard. For one result, see §3. For algorithm 1, more can be said. First, it is easy to concoct additional classes by specifying, say, that searching G_0 from the initial solution leads to a u.c. solution in 2 or 3 steps, whose tableau can be made to look like that of the initial solution by rescaling. After rescaling, we specify that the corresponding M lies in one of the classes for which Lemke's algorithm works. This kind of example is very artificial. Other classes are mentioned in §3, but these are very restrictive.

Part (b) of Theorem 1 suggests a possible approach. Obviously, algorithm 1 will find a solution (if one exists) whenever G is connected (or rather

it has just one component with edges; isolated nodes may exist). G can be obtained as follows: take the 1-skeleton of the polytope given by (4) (i.e. the graph of all extreme points, with adjacency defined obviously). From this, omit all edges lying in both $z_0 = 0$ and $w_0 = 0$, or both $z_1 = 0$ and $w_1 = 0$, etc. The resulting graph is G.

3. Some Classes of Matrices

In this section we show the applicability of algorithms 1 and 2 to some very restrictive classes of matrices. We will use repeatedly the following result of Fiedler and Ptak [6]:

Theorem 2. If the square matrix A has non-positive off diagonal entries, then the following are equivalent:

a) $\exists$ $x > 0$ such that $Ax > 0$

b) A has positive principal minors

c) A^{-1} exists and is non-negative.

Note that (a) can be replaced by $\exists$ $y > 0$ such that $yA > 0$, for (b) holds for A iff it holds for A^T.

We will use also the following lemma identifying undesirable complementary solutions to (4). Let E denote an $n \times n$ matrix of 1's, and for any subsets I,J of $\{1,2,\ldots n\}$, and any $n \times n$ matrix A, let A_{IJ} denote the $|I| \times |J|$ submatrix of A whose rows (columns) are indexed by I(J). Similarly we define x_I for a $n \times 1$ or $1 \times n$ vector. Then we have:

Lemma 1. Let I,J be a partition of $\{1,2,\ldots,n\}$. Then if $(E-M)_{II}$ is non singular, the following is a u.c. solution to (4), ignoring non-negativity:

$$w_0 = 0 \qquad z_0 = k - e_I^T(E_{II}-M_{II})^{-1}(ke_I+q_I)$$

$$w_I = 0 \qquad z_I = (E_{II}-M_{II})^{-1}(ke_I+q_I)$$

$$w_J = ke_J + q_J - (E_{JI}-M_{JI})(E_{II}-M_{II})^{-1}(ke_I+q_I) \qquad z_J = 0.$$

It is feasible if

$$(E_{II}-M_{II})^{-1}e_I > 0, \tag{9}$$

$$e_I^T(E_{II}-M_{II})^{-1}e_I < 1, \tag{10}$$

and

$$(E_{JI}-M_{JI})(E_{II}-M_{II})^{-1}e_I < e_J; \tag{11}$$

and infeasible if (9),(10) and (11), with equality permitted, are not all true.

Proof. Substitution confirms the first part. Conditions (9), (10) and (11) imply that the coefficients of k in z_0, z_I and w_J are strictly positive. Let k' be the value of k so that z_0, z_I and w_J are all non-negative, and one is zero. Then we have a basic solution to (3), and so by our assumption on k, $k > k'$, and the solution is feasible. Now assume, say, $e_I^T(E_{II}-M_{II})^{-1}e_I > 1$, and there is some k for which the solution is feasible. Increasing k, z_0 decreases, so there is a k' giving a basic solution to (3). $k > k'$, so the solution is infeasible. The argument is similar for other "strong" violations of (9), (10) and (11).

We now give a class of matrices M for which Lemke's algorithm (and

hence algorithms 1 and 2) always finds a solution (cf. the discussion of M with $m_{ij} \leq 0$ in [11]):

Theorem 3. Suppose M satisfies $m_{ij} \leq 0$ for $i \neq j$ and $\exists\ v > 0$ such that $vM > 0$, then if $\lambda > \frac{1}{(vM)_j}ve$ for all j, Lemke's algorithm applied to λM will find a solution to the LCP.

Proof. Note that multiplying M by λ is just equivalent to scaling the z-variables. Lemke's algorithm, in the form of Step 1 of algorithms 1 and 2, will either find a d.c. solution or a u.c. solution with basic variables, say, z_0, z_I and w_J for some partition $I \neq \emptyset, J$. We will show that the latter cannot occur.

$\lambda M_{II} - E_{II}$ has non-positive off-diagonal entries, and $v_I(\lambda M_{II} - E_{II}) > (\lambda vM - vE)_I > 0$, so by Theorem 2 $\lambda M_{II} - E_{II}$ is invertible with non-negative inverse. This implies that $(E_{II} - \lambda M_{II})^{-1} e_I < 0$, and so by Lemma 1, the basic z_0, z_I and w_J is infeasible. Thus no u.c. solution exists with $I \neq \emptyset$, and hence Lemke's algorithm will find a solution to the LCP.

Our second result is based on Saigal's construction of a class of matrices on which Lemke's algorithm fails; we show that our algorithms will process these problems.

Theorem 4. Suppose M satisfies $m_{ij} > 0$ for $i \neq j$ and $\exists\ v > 0$ such that $vM < 0$, then if $\lambda > -\frac{1}{(vM)_j}ve$ for all j, and $\lambda > \frac{1}{m_{ij}}$ for all $i \neq j$, algorithm 2 (and hence algorithm 1) applied to λM will process the LCP.

Proof. By Saigal's theorem 3.4 [12], there is a solution to the LCP only if $-M^{-1}q \geq 0$. We will show that under this condition algorithm 2 will find a

solution to the LCP.

Define $(\bar{w}^0, \bar{z}^0)$ by $w_0^0 = k + e^T M^{-1} q$, $w^0 = 0$, $z_0^0 = 0$, and $z^0 = -M^{-1}q$. For $j = 1,2...n$, let $(\bar{w}^j, \bar{z}^j)$ be the solution given in Lemma 1, where $I = \{1,2...n-j+1\}$. By assumption, $(\bar{w}^0, \bar{z}^0)$ is non-negative and hence feasible. We now show the same for $(\bar{w}^j, \bar{z}^j)$, $j = 1,2...n$.

Now $E_{II} - \lambda M_{II}$ has non-positive off-diagonal entries, and $v_I(E_{II}-\lambda M_{II}) > -\lambda v_I M_{II} > -\lambda(vM)_I > 0$. So by Theorem 2, $(E_{II}-\lambda M_{II})^{-1}$ exists and is non-negative. This proves (9).

Furthermore, $e_I^T(E_{II}-\lambda M_{II})^{-1}e_I < \frac{1}{v_I e_I}(-\lambda v_I M_{II})(E_{II}-\lambda M_{II})^{-1}e_I$ $< \frac{1}{v_I e_I}(v_I E_{II}-\lambda v_I M_{II})(E_{II}-\lambda M_{II})^{-1}e_I = 1$, proving (10). The first inequality follows from $(E_{II}-\lambda M_{II})^{-1}e_I > 0$, and $v_I E_{II} < (vE)_I < -\lambda(vM)_I < -\lambda v_I M_{II}$.

Finally, condition (11) follows from (9) and (10), since $\lambda M_{IJ} > 0$.

Thus all $(\bar{w}^j, \bar{z}^j)$ are feasible. The following paths in $G_0, G_1 \ldots G_n, G_0$ are easily checked -- all are in fact 1-edge paths.

$$(\bar{w}^0, \bar{z}^0) \xrightarrow[G_0]{} (\bar{w}^1, \bar{z}^1) \xrightarrow[G_1]{} \cdots (\bar{w}^j, \bar{z}^j) \xrightarrow[G_j]{} (\bar{w}^{j+1}, \bar{z}^{j+1}) \longrightarrow \cdots (\bar{w}^n, \bar{z}^n) \xrightarrow[G_n]{} (\bar{w}^*, \bar{z}^*),$$

where $\bar{w}^*, \bar{z}^*$ is the initial solution given by (4).

However, algorithm 2 fails only when an n-path takes one to $(\bar{w}^*, \bar{z}^*)$, and tracing the path backwards, the solution $(\bar{w}^0, \bar{z}^0)$ must have been encountered; since this is a d.c. solution, algorithm 2 would have terminated at this point. Thus algorithm 2 will always find a solution to the LCP if one exists.

Saigal's theorem 3.5 [12] shows that Lemke's algorithm fails on this problem.

Errata Sheet for

COMPLEMENTARITY ALGORITHMS WITHOUT RAYS

by Michael J. Todd

Technical Report No. 195
Department of Operations Research
Cornell University, Ithaca, N.Y.

Page 12. Lines 2,3,4 should read:

Define $(\bar{w}^0,\bar{z}^0)$ by $w_0^0 = k + e^T\lambda^{-1}M^{-1}q$, $w^0 = 0$, $z_0^0 = 0$, and $z^0 = -\lambda^{-1}M^{-1}q$. For $j = 1,2,\ldots,n$, let $(\bar{w}^j,\bar{z}^j)$ be the solution given in Lemma 1 with λM replacing M, where $I = \{1,2,\ldots,n-j+1\}$. By assumption, $(\bar{w}^0,\bar{z}^0)$ is non-negative and hence feasible. We now show the same for $(\bar{w}^j,\bar{z}^j)$, $j = 1,2,\ldots,n$.

Page 12. Lines 13-14 should read:

Thus all $(\bar{w}^j,\bar{z}^j)$ are feasible. The following paths in $G_0,G_1,\ldots,G_n$ are easily checked -- all are in fact 1-edge paths.

4. The Bimatrix Game Reformulation

A bimatrix game is given by two $m \times n$ matrices A and B. Two players each have a finite set of pure strategies; if I plays i and II plays j, I gets a_{ij} and II b_{ij} for $i = 1,2,\ldots m$, $j = 1,2,\ldots n$. Note that A and B are the matrices of payoffs not losses. The set of all possibly mixed strategies for I is $X = \{x \in R^m | e^T x = 1, x \geq 0\}$ and for II is $Y = \{y \in R^n | e^T y = 1, y \geq 0\}$. A Nash equilibrium point (NEP) is a pair $(\bar{x},\bar{y})$ satisfying $\bar{x} \in X$, $\bar{y} \in Y$, and for all $x \in X$, $y \in Y$, $x^T A\bar{y} \leq \bar{x}^T A\bar{y}$ and $\bar{x}^T By \leq \bar{x}^T B\bar{y}$.

It can be immediately checked that if $(\bar{x},\bar{y})$ is an NEP to the game with payoff matrices A,B, it is also an NEP to the game with payoff matrices A',B', where A' differs from A, and B' from B, only by a constant in all entries. Thus we can assume without loss of generality that A and B are both positive, both negative, etc.

The original algorithm of Lemke and Howson [8] to find an NEP to the bimatrix game used the fact that an NEP can be found from a solution to:

$$\begin{aligned} &(-I,-A)u = e, \quad u \geq 0 \\ &\qquad\qquad\qquad\qquad \text{with} \quad u^T v = 0, \\ &(-B^T,-I)v = e, \quad v \geq 0 \end{aligned} \tag{12}$$

where A and B are supposed negative. An NEP can also be found from a solution to:

$$\begin{aligned} &(I,A)u = e, \quad u \geq 0 \\ &\qquad\qquad\qquad\qquad \text{with} \quad u^T v = 0, \\ &(-B^T,-I)v = e, \quad v \geq 0 \end{aligned} \tag{13}$$

where now we suppose $A > 0$, $B < 0$. See, e.g., [5].

In both these schemes, there is a one-one correspondence between NEP's and solutions u,v. Complementary pivot algorithms can be applied to both (12) and (13). In (12), both systems are unbounded, and two initial pivots are necessary to obtain feasibility. In (13), the second system is unbounded, and one initial pivot is necessary to obtain feasibility.

We obtain a third scheme which can be used to find an NEP. Both systems will be bounded, and no initial pivots are necessary. The price we pay is that the one-one correspondence of the schemes above is destroyed.

For $t \in R^{m+n}$, let t^1 denote $(t_1,t_2,\ldots t_m)^T$ and t^2 denote $(t_{m+1},t_{m+2},\ldots t_{m+n})^T$. Then we have:

Theorem 5. Assume $A,B > 0$. Then

a) If $(\bar{x},\bar{y})$ is an NEP to the bimatrix game with payoff matrices A,B, then there is a solution to

$$(14) \qquad \begin{aligned} (I,A)u &= e, \quad u \geq 0, \\ u^T v &= 0, \\ (B^T,I)v &= e, \quad v \geq 0, \end{aligned}$$

with $u^2 = \frac{1}{\bar{x}^T A \bar{y}} \bar{y}$ and $v^1 = \frac{1}{\bar{x}^T B \bar{y}} \bar{x}$.

b) If u,v is a solution to (14), other than $u^1 = e$, $u^2 = 0$, $v^1 = 0$, $v^2 = e$, then $(\bar{x},\bar{y})$ is an NEP to the bimatrix game with payoff matrices A,B, where $\bar{x} = \frac{1}{e^T v^1} v^1$, $\bar{y} = \frac{1}{e^T u^2} u^2$.

The proof of Theorem 5 follows analogously to those for schemes (12) and (13) and will be omitted.

Note the exceptional case in (b); there is just one solution to (14) which does not correspond to an NEP.

Define complementary solutions, i-almost complementary solutions to (14) and G_i in the obvious way, for $i = 1,2,...m+n$. Then (14) gives the algorithm: pick some i from $1,2,...m+n$. Then search G_i from the unique undesirable complementary solution given in (b) of Theorem 5 to find another solution to (14), hence an NEP.

This formulation has the following advantages: firstly, since both systems are bounded, no arguments need be made that all pivots encountered can in fact be completed -- this is true for all pivots in (14). Secondly, i can be chosen freely from $1,2,...m+n$, thus taking advantage of any knowledge of strategies used in an NEP.

Discussion

Many complementary pivot algorithms deal with "incomplete" pivoting structures, where not all pivots can be made. There are two general techniques for dealing with this:

a) work throughout with incomplete pivoting structures, by defining rays, boundaries, etc. This is the approach of many particular applications, and the general theory of [7].

b) artificially complete the pivoting structures, adding vertices if necessary. This is the approach of [14] and the present paper.

The obvious argument in favor of (a) is that the original problem is not distorted, as in (b). Much research has been concerned with finding conditions for particular applications under which no rays or boundaries will be encountered, and the algorithm will produce a valid solution.

On the other hand, technique (b) has the advantage that a general theory of complementary pivoting can be based on the simple notion of complete pivoting structure, as in [14]. The present paper shows another advantage. Although undesirable solutions may be added by the artificial completion, these when found (which indicates failure in technique (a)) allow new paths to be searched to find desirable solutions. A more powerful general algorithm results.

References

[1] Beale, E.M.L., "Numerical Methods," Nonlinear Programming, J. Abadie, (Ed), North Holland, Amsterdam, 1967.

[2] Cottle, R.W., and G.B. Dantzig, "Complementary Pivot Thecry of Mathematical Programming," Lin. Alg. and its Appns., Vol. 1, 1968.

[3] _____, "A Generalization of the Linear Complementarity Problem," J. of Comb. Thy., Vol. 8, No. 1, 1970.

[4] Dantzig, G.B., and R.W. Cottle, "Positive (Semi-)Definite Programming," Nonlinear Programming, J. Abadie, (Ed.), North Holland, Amsterdam, 1967.

[5] Eaves, B.C., "The Linear Complementarity Problem," Management Science, Vol. 17, No. 9, 1971.

[6] Fiedler, M., and V. Ptak, "On Matrices with Non-positive Off-diagonal Elements and Positive Principal Minors," Czech.Math. J., Vol. 12, 1962.

[7] Gould, F.J., and J.W. Tolle, "A Unified Approach to Complementarity in Opimization," Report No. 7312, Center for Mathematical Studies in Business and Economics, University of Chicago, February 1973.

[8] Lemke, C.E., and J.T. Howson, Jr., "Equilibrium Points of Bimatrix Games," J. SIAM, Vol. 12, No. 2, 1964.

[9] Lemke, C.E., "Bimatrix Equilibrium Points and Mathematical Programming," Management Science, Vol. 11, No. 7, 1965.

[10] _____, "On Complementary Pivot Theory," Mathematics of the Decision Sciences, G.B. Dantzig and A.F. Veinott, Jr., (Eds.), Amer. Math. Soc., 1968.

[11] _____, "Recent Results on Complementarity Problems," Nonlinear Programming, J.B. Rosen, O.L. Mangasarian, and K. Ritter, (Eds.), Academic Press, New York, 1970.

[12] Saigal, R., "On the Class of Complementary Cones and Lemke's Algorithm," SIAM J. of Appl. Math., Vol. 23, No. 1, 1972.

[13] Scarf, H., "An Algorithm for a Class of Non-convex Programming Problems," Cowles Foundation Discussion Paper No. 211, Yale University, 1966.

[14] Todd, M.J., "A Generalized Complementary Pivot Algorithm," to appear in Mathematical Programming.