7-1-1966

An Algorithm for a Class of Nonconvex Programming Problems

Herbert E. Scarf

COWLES FOUNDATION FOR RESEARCH IN ECONOMICS

AT YALE UNIVERSITY

Box 2125, Yale Station
New Haven, Connecticut

COWLES FOUNDATION DISCUSSION PAPER NO. 211

An Algorithm for a Class of Nonconvex Programming Problems

Herbert Scarf

July 14, 1966

An Algorithm for a Class of Nonconvex

Programming Problems

by

Herbert Scarf*

In a recent paper [1], Lemke described the following problem: Let

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \\ a_{n1} & \cdots & a_{nn} \end{bmatrix},$$

be a square matrix of size $n \times n$ and $b = (b_1, \ldots, b_n)$ a vector of size n. Under what conditions can we say that the equations

$$\begin{aligned} a_{11}x_1 + \ldots + a_{1n}x_n - y_1 &= b_1 \\ &\vdots \\ a_{n1}x_1 + \ldots + a_{nn}x_n - y_n &= b_n \end{aligned}$$

have a solution in nonnegative variables x and y, with $x_i y_i = 0$ for every index i?

Lemke offers a variety of conditions which guarantee an affirmative answer to this question. For example, such a solution may be found if the matrix A and the vector b are both strictly positive, and in many other cases as well. In addition to these sufficient conditions, a finite algorithm

---

* The research described in this paper was carried out under a grant from the National Science Foundation. I wish to thank Lloyd Shapley who rescued me from a serious misunderstanding at one point in the development of this work.

based on ordinary pivot steps, is given for calculating a solution to the problem.

The formulation of the problem seems artificial but its importance derives from the observation that a number of problems in mathematical programming can be put into this form. For example, a solution of the linear programming problem

$$\min \; c_1x_1 + \dots + c_nx_n$$

$$\text{subject to} \qquad \begin{array}{c} d_{11}x_1 + \dots + d_{1n}x_n \geq e_1 \\ \vdots \\ d_{m1}x_1 + \dots + d_{mn}x_n \geq e_m \end{array}$$

$$\text{and} \quad x_j \geq 0 \; ;$$

may be found by solving Lemke's problem for the matrix

$$A = \left[\begin{array}{c|c} O & \begin{array}{ccc} d_{11} & \dots & d_{1n} \\ \vdots & & \\ d_{m1} & \dots & d_{mn} \end{array} \\ \hline \begin{array}{ccc} -d_{11} & \dots & -d_{m1} \\ \vdots & & \\ -d_{1n} & \dots & -d_{mn} \end{array} & O \end{array} \right] ,$$

and the vector $b = (\epsilon_1, \dots \epsilon_m , - c_1 , \dots - c_n)$. Quadratic programming problems may put in a similar form by replacing the $m \times m$ submatrix of zeros appearing in the upper left hand corner of A by a nonzero symmetric matrix. And finally, Nash equilibrium points for a two person nonzero sum game may be found by the solution of Lemke's problem with the matrix A in the form

$$A = \left[\begin{array}{c|c} \bigcirc & \begin{matrix} c_{11} & \cdots & c_{1n} \\ \vdots & & \\ c_{m1} & \cdots & c_{mn} \end{matrix} \\ \hline \begin{matrix} d_{11} & \cdots & d_{m1} \\ \vdots & & \\ d_{1n} & \cdots & d_{mn} \end{matrix} & \bigcirc \end{array}\right] .$$

Lemke's algorithm is based on conventional pivot steps of the sort used in linear programming, but the proof that the algorithm terminates in a finite number of steps with the desired solution, is thoroughly original and differs from any previously given proof of finiteness for similar problems. The proof may be applied to algorithms in which conventional pivot steps are replaced by alternative constructions. For example, in [2,3] I introduced the notions of an ordinal basis and an ordinal pivot step; and by combining these ideas with Lemke's finiteness proof, was able to develop an algorithm for calculating approximately a point in the core of an n person cooperative game. The algorithm also could be used to prove Brouwer's fixed point theorem. The present note applies Lemke's finiteness proof to a third concept of basis and pivot step. The new algorithm which results may be applied to calculate without any approximation a vector in the core of an exchange economy with piecewise linear indifference curves, and will probably be more efficient than the previous approximate algorithm, in that larger movements are taken on each pivot step.

The problem to which the new algorithm addresses itself is the following: Let

$$\begin{gathered} a_1(x_1, \dots x_n) \\ \vdots \\ a_n(x_1, \dots x_n) , \end{gathered}$$

be n functions, <u>each</u> of which is the <u>maximum</u> of a finite number of homogeneous linear functions, and let $b = (b_1, \dots b_n)$ be a vector of size n. Under what conditions can we say the the equations

$$\begin{gathered} a_1(x_1, \dots x_n) - y_1 = b_1 \\ \vdots \\ a_n(x_1, \dots x_n) - y_n = b_n , \end{gathered}$$

have a solution in nonnegative variables x and y with $x_i y_i = 0$ for all i ?

Of course, the important difference between this problem and Lemke's is that each linear function is now replaced by the <u>maximum</u> of several linear functions. If $a_i(x)$ were the minimum rather than the maximum of linear functions the problem could be solved by a trivial reformulation of Lemke's problem. The use of the maximum takes the problem into the area of integer programming, and it is therefore somewhat surprising that definite existence theorems can be obtained. Moreover this is the correct formulation for calculating a vector in the core of an n person game, in which case $a_i(x)$ is the support function for a certain polyhedral convex set V_i, i.e., $a_i(x) = \sup \{\alpha \mid \alpha x \in V_i\}$.

Before giving a set of sufficient conditions for the existence of a solution to this problem, it will be useful to introduce some notation and definitions. Each $a_i(x)$ is the maximum of a finite number of linear functions, and the combined data of the problem may be represented by a matrix.

$$A = \begin{array}{cc}
\begin{array}{cccc} x_1 & \cdots & x_n & \end{array} & \\
\left[\begin{array}{cccc} a_{11} & \cdots & a_{1n} & b_1 \\ \vdots & & & \\ \hline & & & \\ \hline a_{m1} & \cdots & a_{mn} & b_1 \\ \hline 1 & \cdots & 0 & 0 \\ \hline \vdots & & & \\ \hline 0 & \cdots & 1 & 0 \\ \hline \end{array}\right] & \begin{array}{c} S_1 \\ \vdots \\ S_n \\ S_{-1} \\ \\ S_{-n} \end{array}
\end{array}$$

The first m rows of this matrix are grouped into n sets of rows $S_1, \ldots S_n$, with S_k the set of rows used in forming the function $a_k(x)$. Each of the final n rows appears in a set by itself. The constants b_i should be equal for all i in the same set S_k, but this will never be used in the subsequent part of the paper.

The following notion of a "feasible basis" for this matrix is appropriate for our problem.

DEFINITION: A set of n rows of the matrix A, with indices $i_1, i_2, \ldots i_n$ is defined to be a "feasible basis" if the submatrix

$$\begin{bmatrix} a_{i_1 1} & \cdots & a_{i_1 n} \\ \vdots & & \\ a_{i_n 1} & \cdots & a_{i_n n} \end{bmatrix}$$

is nonsingular, and if the solution to the equations

$\sum_{j=1}^{n} a_{i_\alpha, j}\, x_j = b_{i_\alpha}$ satisfies the following two conditions:

1. If S_k contains a row in the basis, then $\Sigma\, a_{ij} x_j \leq b_i$ for all i in S_k.

2. If S_k contains no row in the basis, then $\Sigma\, a_{ij} x_j \geq b_i$ for <u>at least one</u> i in S_k.

We shall frequently say that if a set S_k contains a row in the basis, then S_k is in the basis, and if S_k contains no rows in the basis, then S_k is not in the basis. Since a given S_k may contain more than one row in the basis, there may be fewer than n sets in the basis.

It should be clear that to solve the problem previously given, it is sufficient to find a feasible basis such that the variable $x_k = 0$ if the set S_k has no rows in the basis, for $k = 1, \ldots n$. In other words the basis should have the property that for every pair S_k and S_{-k} (with $k = 1, \ldots n$) at least one of the pair of sets is in the basis.

Theorem: Let the matrix A and the partition $\{S_k\}$ be given. Assume that

$$\sum_{k=1}^{n} x_k \max_{i \in S_k} \left(\sum_j a_{ij} x_j\right) \leq 0$$

for $x \geqq 0$ implies $x = 0$.

Then there is a feasible basis as defined above, with the property that for every $k = 1, \ldots n$, at least one of the pair S_k, S_{-k} is in the basis.

The hypotheses of the theorem, which are a generalization of those given by Lemke for the case in which each S_k consists of a single row, are by no means necessary for the validity of the conclusion, or for the successful application of the algorithm. For example the conditions are not satisfied if the problem is that of calculating a vector in the core of an n person game, in which case the first m rows of the matrix are given by

$$\begin{bmatrix} & A_1 & \cdot \\ & \cdot & \cdot \\ & \cdot & \cdot \\ 0 & \cdot & \cdot \\ & \cdot & \cdot \\ & A_{2^n-1} & \cdot \\ -D & 0 & -e \end{bmatrix}.$$

The rows of each submatrix A_i are the normals to the supporting hyperplanes for a convex polyhedral set, D is the incidence matrix of players versus coalitions, and e a vector all of whose components are 1. Nevertheless the algorithm is successful for this problem.

The particular version of the algorithm to be discussed here involves following Lemke by introducing an extra column and two extra rows in A, and obtaining a new matrix

$$
\tilde{A} = \begin{array}{c|c}
\begin{array}{ccccccc} x_1 & \cdots & x_n & x_{n+1} & \end{array} & \\
\left[\begin{array}{ccccc}
a_{11} & \cdots & a_{1n} & 1 & b_1 \\
\vdots & & \vdots & \vdots & \vdots \\
\hline
\vdots & & \vdots & \vdots & \vdots \\
\hline
\vdots & & & & \\
a_{m1} & \cdots & a_{mn} & 1 & b_m \\
\hline
-1 & \cdots & -1 & 0 & -M \\
\hline
1 \;\; 0 & \cdots & 0 & 0 & 0 \\
\hline
\vdots & & & & \\
\hline
0 \;\; 0 & \cdots & 0 & 1 & 0
\end{array}\right]
&
\begin{array}{l}
S_1 \\ \\ \vdots \\ \\ S_n \\ S_{n+1} \\ S_{-1} \\ \\ S_{-n-1}
\end{array}
\end{array}
$$

A new variable x_{n+1} has been introduced, and two new rows, one row constituting the set S_{n+1} and another the set S_{-n-1}. The constant M is selected as a large positive number. Of course, a feasible basis for this matrix consists of a set of $n+1$ rows rather than n.

The algorithm will be applied to this matrix rather than the original one, and it will terminate with a feasible basis with the property that for every $k = 1, \dots n+1$ either S_k or S_{-k} is in the basis. In other words we shall obtain a vector $x_1, \dots x_{n+1} \geq 0$, such that for every set $k = 1, \dots n$.

1. $\sum_1^n a_{ij}x_j + x_{n+1} \geq b_i$ for at least one $i \in S_k$, and
2. if $x_k > 0$, then
   $\sum_1^n a_{ij}x_j + x_{n+1} \leq b_i$ for $\underline{\text{all}}$ $i \in S_k$.

Moreover if $x_{n+1} > 0$, then $\sum_1^n x_j = M$.

In order to show that this solution works for the original matrix A, we need only show that $x_{n+1} = 0$, and this is true if $\sum_1^n x_j < M$. It is therefore sufficient to show that M may be selected so large that if 1 and 2 hold then necessarily $\sum_1^n x_j < M$.

Conditions 1 and 2 imply that

$$x_k \sum_1^n a_{ij}x_j + x_k x_{n+1} \leq b_i x_k \quad \text{and therefore}$$

$$x_k \sum_1^n a_{ij}x_j \leq b_i x_k \quad \text{for all } i \in S_k \text{ and all } k = 1, \dots n .$$

It follows that

$$\sum_{k=1}^{n} x_k \max_{i\in S_k} (\Sigma a_{ij}x_j) \leq \sum_{k=1}^{n} \left(\max_{i\in S_k} b_i\right) x_k .$$

But it is impossible to find such vectors which are nonnegative and with $\sum_1^n x_j = M$ becoming arbitrarily large, for if this were so we could define $(\xi_1, \ldots \xi_n)$ to be a limit point of $\left(\frac{x_1}{M}, \ldots \frac{x_n}{M}\right)$. But then $\xi_j \geq 0$, $\sum_1^n \xi_j = 1$, and $\sum_{k=1}^{n} \xi_k \max_{i\in S_k} (\Sigma a_{ij}\xi_j) \leq 0$, which contradicts the hypothesis of our theorem.

This argument permits us to concentrate on the matrix $\tilde{A}$, which is considerably easier to work with than A. There are some problems however in which the matrix A has sufficient properties so that an extension to $\tilde{A}$ is not required, or in which an $\tilde{A}$ different from the above is more useful.

It is convenient to make the following nondegeneracy assumption, which can easily be brought about by a perturbation of the b's.

NONDEGENERACY ASSUMPTION: Let x satisfy the equations

$$\sum_1^{n+1} a_{i_\alpha, j}x_j = b_{i_\alpha}, \text{ where the indices } i_\alpha \text{ range over}$$

any $(n+1)$ rows of $\tilde{A}$. Then for any row i different from one of these, we have

$$\Sigma a_{ij}x_j \neq b_i .$$

We shall begin the algorithm with a specific feasible basis consisting of $S_{-1}, \ldots S_{-n}$, and some set S_{k*} selected from among the first n sets of $\tilde{A}$. For each $k = 1, \ldots n$ calculate

$$\min_{i \in S_k} b_i \, .$$

Then select the set k so as to maximize this quantity. By the nondegeneracy assumption these will be a unique set S_{k*} which maximizes, and a unique row $i*$ in that set which gives the minimum ratio. The first n rows of the unit matrix of $\tilde{A}$ plus the row $i*$ may easily be shown to be a feasible basis for $\tilde{A}$ if $b_{i*} > 0$. If $b_{i*} \leq 0$, then the solution to the problem is given by the zero vector.

Our basis has the property that for every k other than $k = n + 1$ at least one of the pair S_k, S_{-k} is in the basis. The algorithm will involve only bases of this sort. In other words at every step we will have a basis including possibly some of the last $n + 1$ rows; neither S_{n+1} nor S_{-n-1} will be in the basis, but for every other pair S_k, S_{-k} at least one will be in the basis.

It should be clear that at least n of the $2(n+1)$ sets must appear in such a basis. If in fact $(n+1)$ of the $2(n+1)$ sets appear in the basis, then each of these sets must contain precisely one of the $(n+1)$ rows appearing in the basis. A basis of this sort will be said to be of type 1. If n distinct sets appear in the basis, then one of them will contain two rows and the others precisely one row. A basis of this sort will be said to be of type 2. These considerations are of sufficient importance for the algorithm, to be summarized in the following definition.

DEFINITION: A basis is of type 1 if $n + 1$ distinct sets appear in the basis. The basis is of type 2 if n distinct sets appear in the basis.

Of course, there are feasible bases other than those of type 1 or 2, but the algorithm will never be at such a feasible basis if for every k other than $k = n + 1$ at least one of the pair S_k, S_{-k} is in the basis.

The algorithm will take us systematically from one such feasible basis to another. As we shall see, the appropriate concept of a "pivot step" is to select an arbitrary row appearing in a basis and to remove it, with a new row taking its place. If the basis is of the sort described above then there will be <u>two</u> natural pivot steps to take which preserve the form of the basis. For example if the basis is of type 1, then there is precisely one index k, such that <u>both</u> S_k and S_{-k} are in the basis. Each of these sets contains one row in the basis. Either of these rows may be removed giving rise to <u>two</u> pivot steps.

On the other hand, if the basis is of type 2, then there is precisely one set which contains two rows in the basis, and either of these two rows may be removed.

The algorithm proceeds as follows. At each stage there will be two natural pivot steps to take. <u>One</u> of these will have been taken in order to reach the point where we are, so that the <u>other</u> pivot step should be taken. As we shall see, both of these pivot steps may be carried out if we are not at the initial basis, where only one can be implemented. Moreover, the algorithm can not return to a basis already passed, since, as we shall see, pivot steps are reversible. Since there are only a finite number of feasible bases, the algorithm must terminate in a finite number of steps with a solution to the problem.

This argument depends, of course, on exhibiting the details of a pivot step appropriate to our notion of a feasible basis.

1. The Details of a Pivot Step From a Type 1 Basis.

Let $(i_1, \ldots i_{n+1})$ be the rows of a type 1 basis. The row i_α is contained in the set S_{k_α}, and these (n+1) sets are distinct. $x^o = (x_1^o, \ldots x_{n+1}^o)$ will represent the solution of the equations

$$\Sigma\, a_{i_\alpha, j}\, x_j = b_{i_\alpha} ,$$

and the following properties are assumed to hold:

1. If S_k is in the basis, then

$$\Sigma\, a_{ij} x_j^o < b_i \quad \text{for all} \quad i \text{ in } S_k \text{ other}$$

than the row in S_k which is in the basis.

2. If S_k is not in the basis, then

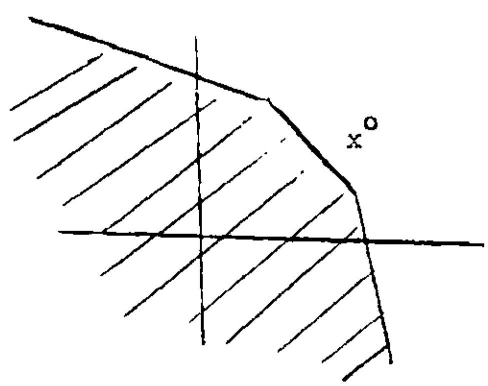
$$\Sigma\, a_{ij} x_j^o > b_i \quad \text{for at least one} \quad i \text{ in } S_k .$$

(The strict inequalities are used here because of the nondegeneracy assumption.)

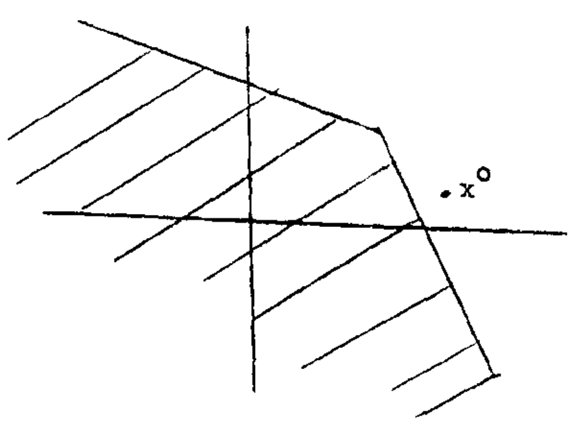
There is a simple geometric interpretation for a type 1 basis. If we define, for each k ,

$$C_k = \{ x \mid \sum_j a_{ij} x_j \leq b_i \quad \text{for all } i \text{ in } S_k \} ,$$

then C_k is a polyhedral convex set in $(n+1)$ dimensional space. If S_k is in the basis, then x^0 lies on the boundary of C_k , and is in only one of the defining hyperplanes



If S_k is not in the basis, then x^0 lies outside of C_k .



A particular row i^* in a particular set S_{k^*}, will have been selected to be removed from the basis. To do this we look for solutions of the equations

$$\Sigma\, a_{i_\alpha, j}\, x_j = b_{i_\alpha} \quad \text{for all } i_\alpha \text{ in the basis, other}$$

than i^*, and

$$\Sigma\, a_{i^*, j}\, x_j = b_{i^*} + \epsilon \,, \quad \text{with } \epsilon > 0 \,.$$

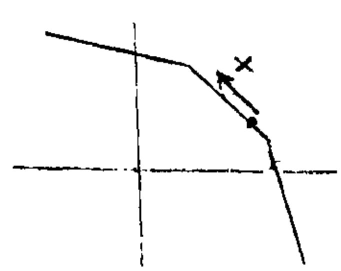
This gives rise to a one parameter family of solutions

$$x_j = x_j^0 + \epsilon \xi_j \quad \text{where}$$

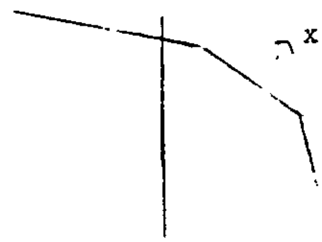
$$\Sigma\, a_{i_\alpha, j} \xi_j = 0 \quad \text{for all } i_\alpha \text{ in the basis other than } i^* \,, \text{ and}$$

$$\Sigma\, a_{i^*, j} \xi_j = 1 \,.$$

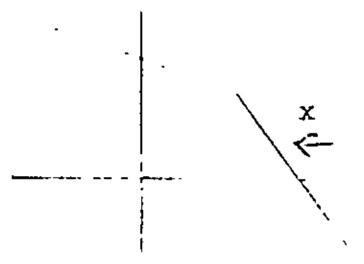
As ϵ increases from zero, the vector x moves in a straight line. If S_k is in the basis, and not equal to k^*, then x remains in one bounding hyperplane associated with C_k, and may or may not eventually reach another bounding hyperplane.



If $k = k^*$, then the vector x departs from C_k.



If S_k is not in the basis, then x moves along a straight line which may or may not eventually intersect C_k.



Every set S_k which is not in the basis is examined to determine the smallest nonnegative value of ϵ for which x is in C_k, if there is such a value of ϵ, i.e., the smallest value of ϵ for which

$$\Sigma\, a_{ij} x_j^o + \epsilon\, \Sigma\, a_{ij} \xi_j \leq b_i \quad \text{for all} \quad i \in S_k .$$

This value of ϵ may be determined as follows, remembering that $\Sigma\, a_{ij} x_j^o - b_i \neq 0$ for any $i \in S_k$, because of the nondegeneracy assumption.

RULE 1. If $\Sigma\, a_{ij} x_j^o - b_i > 0$ and $\Sigma\, a_{ij}\, \epsilon_j \geq 0$, for some $i \in S_k$, then no value of ϵ will work and we skip this set.

RULE 2. If Rule 1 is not invoked for this set, then we calculate

$$\epsilon_k = \max_{\substack{\Sigma a_{ij}x_j^o - b_i > 0 \\ \Sigma a_{ij}\xi_j < 0}} \left(\frac{\Sigma a_{ij}x_j^o - b_i}{-\Sigma a_{ij}\xi_j} \right),$$

and

$$\delta_k = \min_{\substack{\Sigma a_{ij}x_j^o - b_i < 0 \\ \Sigma a_{ij}\xi_j > 0}} \left(\frac{\Sigma a_{ij}x_j^o - b_i}{-\Sigma a_{ij}\xi_j} \right).$$

δ_k is taken to be ∞ if there are no rows in S_k meeting the conditions of its definition. If Rule 1 does not apply, then there will be some rows used in defining ϵ_k.

RULE 3. If $\delta_k < \epsilon_k$, skip this set, since no value of ϵ will take th vector x into the set C_k. On the other hand if $\epsilon_k \leq \delta_k$, then x enters the set C_k for $\epsilon = \epsilon_k$, and S_k is a potential set to be brought into the basis when S_{k^*} is removed.

RULE 4. Let $\bar{\epsilon}$ be the minimum of the ϵ_k for all sets S_k which are not in the basis, and which have not been excluded by the applications of the first three rules. If all sets are excluded put $\bar{\epsilon}$ equal to ∞.

We have determined the smallest value of ϵ such that x enters a set C_k with S_k not in the basis. If the pivot step were to take us to another type 1 basis, then the unique row associated with $\bar{\epsilon}$ would be taken into the basis. On the other hand it is possible that a pivot step will require a transition to a type 2 basis. This means that the vector x will reach a second bounding hyperplane for some C_k with S_k in the basis $(k \neq k^*)$, for ϵ smaller than $\bar{\epsilon}$. We determine whether this is true by examining all sets S_k in the basis, other than $k = k^*$, and applying the following rules, realizing that for such sets

$$\Sigma\, a_{ij} x_j^o - b_i \leq 0 \quad \text{for all} \quad i \,.$$

RULE 5. For each S_k in the basis, other than k^*, we calculate

$$\delta_k = \min_{\substack{\Sigma\, a_{ij} x_j^o - b_i < 0 \\ \Sigma\, a_{ij} \xi_j > 0}} \left(\frac{\Sigma\, a_{ij} x_j^o - b_i}{-\Sigma\, a_{ij} \xi_j} \right) ,$$

or ∞ if there are no contenders.

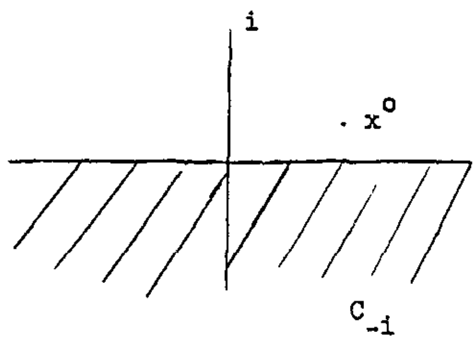
RULE 6. Let $\bar{\delta}$ be the minimum of the δ_k for all sets S_k in the basis other than k^* . If $\bar{\delta} < \bar{\epsilon}$ the pivot step will be to a type 2 basis with the row to be brought in corresponding to that used in defining $\bar{\delta}$. If $\bar{\epsilon} < \bar{\delta}$ (there will be no ties) then the pivot step is to a type 1 basis with the row to be brought in corresponding to that used in defining $\bar{\epsilon}$.

Lemma 1. If neither S_{n+1} nor S_{-n-1} are in a type 1 basis, and for every other pair S_k , S_{-k} at least one is in the basis, then the pivot step can always be carried out, unless the basis consists of

$$S_{k*} ,\ S_{-1} , \ldots, S_{-n} ,\ \text{and we are}$$

attempting to remove S_{k*} .

If a pivot step cannot be carried out it means that the vector x cannot enter any of the sets C_k for which S_k is not in the basis. Assume that a set S_{-i} with $i = 1, \ldots n+1$ is not in the basis. This means that $\Sigma\, a_{ij}x_j^o > b_i$ for this row or $x_i^o > 0$. In order for x not to enter C_{-i}



Rule 1 must be invoked so that $\Sigma\, a_{ij}\xi_j \geq 0$ or $\xi_i \geq 0$.

On the other hand if S_{-i} is in the basis, then either $\xi_i = 1$, if S_{-i} <u>is</u> being removed from the basis, or else $\xi_i = 0$. In any event, for a pivot step not to be carried out we must have $\xi \geq 0$.

Will the vector x eventually enter the set C_{n+1}? Applying Rule 1, we see that for x not to enter C_{n+1} , we must have $\sum_{j=1}^{n} \xi_j \leq 0$, so that for a pivot step not to be carried out we must have $\xi_1 = \ldots = \xi_n = 0$, and $\xi_{n+1} > 0$. But then if S_k with $k = 1, \ldots n$ is <u>any</u> set in the basis, and i the row in S_k which is in the basis,

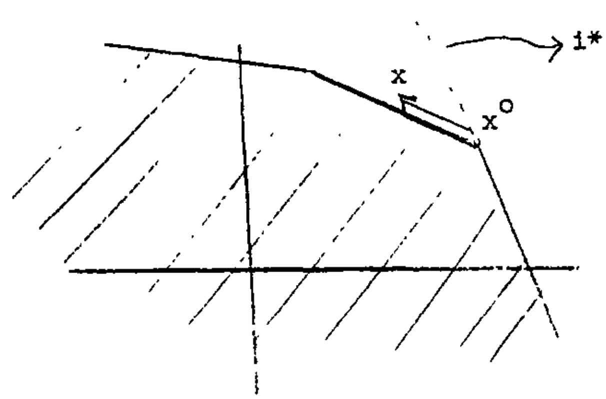
$$\sum_{1}^{n} a_{ij}\xi_j + \xi_{n+1} > 0 \ ,$$

which can only occur if we are removing this set from the basis. Therefore there can only be one set S_k with $k = 1, \dots n$ in the basis, and the basis must consist of $S_1, \dots S_n$ and S_{k*}, with S_{k*} being removed, for the pivot step to be impossible to carry out.

2. The Details of a Pivot Step From a Type 2 Basis.

A type 2 basis consists of n rather than $n + 1$ sets $S_{k_1}, \dots S_{k_n}$. The sets S_{n+1} and S_{-n-1} are not in the basis, and for every other pair S_k and S_{-k} $(k = 1, \dots n)$ precisely one is in the basis and the other not. One of these sets S_{k*} has two rows in the basis, and one of these two rows is to be removed.

If we define the sets C_k as before, then the diagrams of the previous section are applicable except for the set C_{k*}, which has the property that x^o lies on the intersection of two bounding faces.



One of these two bounding faces, say the one corresponding to the row $i*$ of $\tilde{A}$ is to be removed from the basis. We therefore look for solutions to the equations

$$\Sigma\, a_{i_\alpha, j}\, x_j = b_{i_\alpha} \quad \text{for all} \quad i_\alpha \text{ in the}$$

basis other than $i*$, and

$$\Sigma\, a_{i*, j} x_j = b_{i*} - \epsilon\ , \quad \text{with} \quad \epsilon > 0\ .$$

The reason for the $-\epsilon$ instead of $+\epsilon$ is that in the pivot step from a type 2 basis the vector moves "inside" the face determined by row $i*$, rather than "outside."

The vector x is given by $x_j = x_j^o + \epsilon \xi_j$ where

$$\Sigma\, a_{i_\alpha, j} \xi_j = 0 \quad \text{for all} \quad i_\alpha \quad \text{in the basis other than} \quad i* , \quad \text{and}$$

$$\Sigma\, a_{i*, j} \xi_j = -1\ .$$

As in the pivot step from a type 1 basis we begin by examining the sets S_k not in the basis, to determine the smallest nonnegative value of ϵ , if there is such a value of ϵ , so that x is in C_k, i.e., the smallest value of ϵ such that

$$\Sigma\, a_{ij} x_j^o + \epsilon\, \Sigma\, a_{ij} \xi_j \leq b_i \quad \text{for all} \quad i \in S_k\ .$$

The first four RULES for a type one pivot step may be applied without change to determine $\bar{\epsilon}$, the smallest value of ϵ such that x enters one of the sets S_k with S_k not in the basis. If the transition is to a type 1 basis, the unique row associated with $\bar{\epsilon}$ is taken into the basis. On the other hand the transition may be to a type 2 basis, in which case x will reach a second bounding hyperplane for some C_k with S_k in the basis (including possibly C_{k*}) for ϵ smaller than $\bar{\epsilon}$. Rules 5 and 6 are therefore applied with the modification that all sets S_k in the basis are examined.

As we see the only differences between a type 1 and type 2 pivot are that for the latter

$$\Sigma\, a_{i*,j}\xi_j = -1 \ , \quad \text{and}$$

that all sets in the basis are examined in Rules 5 and 6.

Lemma 2. If neither S_{n+1} nor S_{-n-1} are in a type 2 basis, and precisely one of every other pair is in the basis, with S_{k*} having two rows in the basis, then the indicated pivot step can always be carried out.

The argument is similar to that given for Lemma 1. If the pivot step cannot be carried out then x cannot enter C_{-i} if S_{-i} is not in the basis, and therefore $\xi_i \geq 0$. On the other hand if S_{-i} is in the basis, then since it has only one row it cannot be the set S_{k*}, and we must have $\xi_i = 0$. If a pivot step cannot be carried out, then $\xi_i \geq 0$ for all i.

As before, we ask whether x will eventually enter C_{n+1}, since S_{n+1} is not in the basis. Applying Rule 1 again we see that

$\sum_{1}^{n} \xi_j \leq 0$, so that $\xi_1 = \ldots = \xi_n = 0$ and $\xi_{n+1} > 0$. But ξ satisfies the equation

$$\sum_{1}^{n} a_{i*,j}\xi_j + \xi_{n+1} = - 1 ,$$

since S_{k*} must be one of the first n sets in order to have at least two rows in it. This latter equation is impossible and this concludes the proof of Lemma 2.

The reader should be able to convince himself, by meditation, that the reverse of a pivot step is also a pivot step. The algorithm may therefore be applied as described above, and our theorem has been demonstrated.

REFERENCES

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

[2] Scarf, H., "The Core of an N Person Game," to be published in Econometrica, 1967.

[3] Scarf, H., "Equilibrium Points of Two Person Games and the Brouwer Fixed Point Theorem," unpublished manuscript.