CBM
R
7626
1985
200

* C I N 0 0 3 7 5 *

FEW

subfaculteit der econometrie

RESEARCH MEMORANDUM

LBURG UNIVERSITY
EPARTMENT OF ECONOMICS
stbus 90153 - 5000 LE Tilburg
therlands

[redacted]

K.U.B.
BIBLIOTHEEK
TILBURG

FEW
200

AN ALGORITHM FOR THE LINEAR COMPLEMENTARITY PROBLEM WITH UPPER AND LOWER BOUNDS

5179

G. VAN DER LAAN[1] AND A.J.J. TALMAN[2]

DECEMBER 1985

1) Department of Actuarial Sciences and Econometrics,
Free University, Amsterdam, The Netherlands.

2) Department of Econometrics, Tilburg University,
Tilburg, The Netherlands.

This research is part of the VF-program "Equilibrium and Disequilibrium in Demand and Supply", which has been approved by the Netherlands Ministry of Education.

AN ALGORITHM FOR THE LINEAR COMPLEMENTARITY PROBLEM WITH UPPER AND LOWER BOUNDS

by

G. VAN DER LAAN and A.J.J. TALMAN.

Abstract.

In this paper the so-called octahedral algorithm for solving systems of nonlinear equations is adapted to solve the linear complementarity problem with upper and lower bounds. The proposed algorithm generates a piecewise linear path from an arbitrarily chosen point z^0 to a solution point. This path is followed by linear programming pivot steps in a system of n linear equations where n is the size of the problem. The starting point z^0 is left in the direction of one of the 2^n vertices of the feasible region, depending on the sign pattern of the function value at z^0. The sign pattern of the linear function and the location of the points in comparison with z^0 completely govern the path of the algorithm. We show that, at least for n=2, the proposed algorithm performs in general better than the generalized Lemke's algorithm.

An algorithm for the linear complementarity problem with upper and lower bounds.

1. Introduction.

The Linear Complementarity Problem (LCP) consists in finding two vectors s and z in R^n such that for given $n\times n$ matrix M and n-vector q

(i) $s = Mz+q$

(ii) $s,z \geq 0$

(iii) $s^T z = 0$.

The LCP is an important problem in mathematical programming (see e.g. Garcia and Gould [2]). Lemke [6] first presented a solution for this problem. Lemke's algorithm is inttialized at $z=0$ and traces from this point a piecewise linear path of points until a solution is obtained or a ray is encountered. Talman and Van der Heyden [9] proposed an algorithm which allows for an arbitrary starting point in the non-negative orthant. When taking the starting point in the interior of this orthant there are 2n different directions to leave this point. Along which direction the starting point is left depends on the component of s having the largest absolute value. In case $z=0$ is chosen to be the starting point the algorithm reduces to Lemke's algorithm.

The feature of allowing arbitrarily chosen starting points has obvious practical merit in such applications as parametric studies or solving a nonlinear complementarity problem through a sequence of approximating LCP's (see e.g. [4] and [7]). In Everts [1] the algorithm of Talman and Van der Heyden was generalized in order to solve the LCP with upper and lower bounds. This Generalized Linear Complementarity Problem (GLCP) consists of finding vectors s and z in R^n such that for given matrix M, n-vector q and n-vectors a and b with $a_j<b_j$, $j=1,\dots,n$,

(i) $s = Mz+q$

(ii) $a \leq z \leq b$

(iii) $z_j = a_j \rightarrow s_j \leq 0$

$a_j < z_j < b_j \rightarrow s_j = 0 \qquad j=1,\dots,n$

$z_j = b_j \rightarrow s_j \geq 0$.

When the path of the algorithm hits a boundary face of the feasible region

$$C^n = \{z \in R^n | a \le z \le b\}$$

the algorithm continues by tracing a path in this face. To each a solution point in a k-face of C^n, $1 \le k \le n$, from an interior point of C^n, the algorithm needs at least 2n-k (pivot) steps and 2n-1 if k=0, i.e. when one of the vertices of C^n is found as a solution point.

In this paper we propose a pivoting algorithm having 2^n rays, each of them leading to a vertex of C^n. The ray along which the algorithm leaves the starting point depends on the sign pattern of the s-value at this point. More generally, the piecewise linear path of the algorithm is determined by the sign pattern of s=Mz+q and the location of z with respect to the starting point. The algorithm is such that a solution point is found as soon as the path of the algorithm hits a face of C^n not containing the starting point. From this we can conclude that if on the starting ray the sign pattern of s does not change the corresponding vertex of C^n solves the GLCP and is found in just one step. In general, if $0 \le k \le n-1$, a solution on a k-face of C^n could be found in k+1 pivot steps, which is considerably less than for the algorithm with 2n rays when k is small with respect to n. This motivates the presentation of the new algorithm. We notice that the algorithm with 2n rays is an adaption of the so-called cubical algorithm for solving a system of nonlinear equations presented in [5] , see also [8]. The new algorithm with 2^n rays is a similar adaption of the octahedral algorithm introduced in Wright [10].

The paper is organized as follows. In section 2 a detailed description of the piecewise linear path followed by the algorithm is given. The steps of the algorithm are presented in section 3. Finally, in section 4 the algorithm is compared with the generalized Lemke's algorithm. Furthermore we present an adaption of the algorithm to solve the classical LCP.

2. Movements of the algorithm.

We consider points (s,z) in $R^n \times C^n$ satisfying

$$s = Mz + q \tag{2.1}$$

where $C^n=\{x\in R^n | a_i \le z_i \le b_i,\ i=1,\dots,n\}$ with $-\infty<a_i<b_i<\infty$ for all i. The case that some of the numbers b_i (a_i) are infinite (minus infinite) is discussed in section 4. Starting at an arbitrary point z^0, the algorithm adjusts z by increasing z_i if $s_i>0$ (and $z_i^0<b_i$) and by decreasing z_i if $s_i<0$ (and $z_i^0>a_i$). More precisely, for each sign vector t in $\{-1,+1\}^n$ a direction $d(t)=v(t)-z^0$ is defined with $v(t)$ the vertex of C^n given by $v_i(t)=b_i$ if $t_i=+1$ and $v_i(t)=a_i$ if $t_i=-1$. Assuming that $s_i^0\neq 0$ for all i, the algorithm leaves z^0 in the direction $d(t^0)$ towards the vertex $v(t^0)$ of C^n with $t^0=\text{sgn } s^0$. So, the direction in which z^0 is left is the direction associated with sgn (s^0). The algorithm leaves this ray as soon as s_i becomes equal to zero for some i, say at the point $\bar{z}$. So for each point z between z^0 and $\bar{z}$ we have that $t = \text{sgn } s = t^0$. Since z lies between $\bar{z}$ and z^0, there is a λ, $0<\lambda\le 1$ such that

$$z_j = z_j^0 + \lambda(b_j-z_j^0) \qquad \text{for all j with } t_j = +1$$

$$z_j = z_j^0 + \lambda(a_j-z_j^0) \qquad \text{for all j with } t_j = -1.$$

The algorithm maintains this property between z_j and t_j as long as $t_j\neq 0$. However, when s_j becomes equal to zero and the associated point z does not solve the problem, then the algorithm continues by decreasing z_j away from $z_j^0+\lambda(b_j-z_j^0)$ if t_j was +1 and increasing z_j away from $z_j^0+\lambda(a_j-z_j^0)$ if t_j was -1, while s_j is kept equal to zero. In general, the algorithm generates a path of points z such that for some λ, $0<\lambda\le 1$,

$$\begin{aligned} z_j &= z_j^0 + \lambda(b_j-z_j^0) && \text{for all j with } t_j = +1\\ z_j &= z_j^0 + \lambda(a_j-z_j^0) && \text{for all j with } t_j = -1 \\ z_j^0 + \lambda(a_j-z_j^0) &\le z_j \le z_j^0 + \lambda(b_j-z_j^0) && \text{for all j with } t_j = 0 \end{aligned} \tag{2.2}$$

where $t = \text{sgn } s$. These complementarity properties between z and s govern the algorithm and define a piecewise linear path of points in C^n connecting z^0 with a solution to the problem. A solution point is reached as soon as $t_j=0$ for all j with $z_j \neq z_j^0$ or $\lambda=1$. In the first case we have that $s_j=0$ if $z_j \neq z_j^0$ while (2.2) implies that $z_j=z_j^0$ iff either $z_j^0=b_j$ and $t_j=+1$ or $z_j^0=a_j$ and $t_j=-1$. If $\lambda=1$ then $z_j=b_j$ if $t_j=+1$, $z_j=a_j$ if $t_j=-1$ and $t_j=0$ if $a_j<z_j<b_j$. In particular we may have that s does not change along the ray $d(t^0)$ when leaving z^0. Then λ becomes 1 at $z=v(t^0)$ and $v(t^0)$ is a solution point. In such a case a solution is found in just one step.

We will prove now that the points (s,z) in $R^n \times C^n$ satisfying (2.2) indeed induce a sequence of adjacent line segments in C^n connecting z^0 and a solution point. Let T be the set of sign vectors in R^n, i.e.

$$T = \{t \in R^n \mid t_i \in \{-1,0,+1\},\ i=1,\dots,n\}.$$

Furthermore, let z^0 be an arbitrary point in C^n which will be the starting point of the algorithm. Then for $t \in T$ we define the convex polyhedral set A(t) by

$$A(t) = \emptyset \quad \text{if } z_i^0 = b_i \text{ for all i with } t_i = +1 \text{ and}$$
$$z_i^0 = a_i \text{ for all i with } t_i = -1,$$

and, otherwise

$$A(t) = \{z \in C^n \mid \begin{array}{ll} z_j=\lambda(b_j-z_j^0)+z_j^0 & \text{if } t_j = +1 \\ z_j=\lambda(a_j-z_j^0)+z_j^0 & \text{if } t_j = -1 \\ \lambda(a_j-z_j^0) \le z_j-z_j^0 \le \lambda(b_j-z_j^0) & \text{if } t_j = 0 \\ \text{with } 0 \le \lambda \le 1 & \end{array} \}. \tag{2.3}$$

For $t \in T$, let $I^-(t)=\{i \in I_n \mid t_i=-1\}$, $I^0(t)=\{i \in I_n \mid t_i=0\}$ and let $I^+(t)=\{i \in I_n \mid t_i=+1\}$. Then

$$\dim A(t) = |I^0(t)| + 1$$

if $A(t)$ is nonempty. If $t \in \{-1,1\}^n$ then dim $A(t)=1$ (unless $z^0=v(t)$) and $A(t)$ is the line segment connecting z^0 with $v(t)$, i.e. $A(t)$ is the ray along which the direction $d(t)=v(t)-z^0$ points at z^0. So, if z^0 is not a vertex of C^n, there are 2^n directions along one of which z^0 is left. When z^0 is a vertex of C^n, there are 2^n-1 directions. The algorithm leaves z^0 along the ray $A(t^0)$ for which $t^0=\text{sgn}\ (s^0)$. Observe that z^0 solves the problem if $z^0=v(t^0)$. In general, the algorithm generates points z in C^n such that for some t in T both z lies in $A(t)$ and $t=\text{sgn}\ s$. For $t \in T$, let $C(t)$ be defined by

$$C(t) = \text{Cl}\{z \in C^n \mid \text{sgn}(Mz+q) = t\}$$

and let $B(t) = C(t) \cap A(t)$. A point z satisfies (2.2) if and only if z lies in $B(t)$ for some $t \in T$. We now introduce basic and nonbasic variables. Notice that z lies in $B(\text{sgn}\ s)$.

<u>Definition 2.1</u>. For some $z \in C^n$, $z \neq z^0$, let $A(\bar{t})$ be the smallest set $A(t)$ containing z in its interior. Then the variable z_j, $j \in I_n$ is said to be nonbasic if $\bar{t}_j \neq 0$. With $s=Mz+q$, the variable s_j is said to be nonbasic if $s_j=0$. Furthermore, let λ be defined as in 2.2 with $t=\bar{t}$. Then λ is said to be nonbasic if $\lambda=1$. Finally, for $z=z^0$, λ is defined to be equal to zero and all variables z_j, $j \in I_n$ and λ are said to be nonbasic. When not nonbasic, a variable is said to be basic.

<u>Definition 2.2</u>. A pair (s,z) is called complementary if for each $j \in I_n$, either or both z_j and s_j are nonbasic.

<u>Nondegeneracy assumption 2.1</u>. For each z in $B(t)$, $t \in T$, holds that among the $2n+1$ variables (z,s,λ) with $s=Mz+q$ and λ as defined in (2.2) ($\lambda=0$ if $z=z^0$) at most $n+1$ variables are nonbasic.

This assumption does not cause a loss of generality, since if degeneracy occurs a slight perturbation of the data (M,q) will restore the assumption.

By definition, the pair (s^0,z^0) is complementary because at z^0 all variables z_j are defined to be nonbasic. Since also λ is nonbasic at z^0 assumption 2.1 implies that $s_j^0 \neq 0$ for all j. Hence, z^0 lies in

$B(t^0)$, $t^0=\text{sgn } s^0$ and in no other set $B(t)$, $t\neq t^0$. The set $B(t^0)$ is obtained by increasing λ from 0 at z^0. Doing so a line segment of points z in $A(t^0)$ is generated while complementarity between z and s is maintained. This movement is pursued until one -and just one, because of assumption 2.1- basic variable becomes nonbasic. At such a point z either λ becomes equal to one or $s_j=0$ for just one $j\in I_n$. In the first case a solution has been reached, as has been shown before. In the latter case the corresponding variable z_j becomes basic and the algorithm moves into the associated region $A(t)$, $t = \text{sgn } s$, tracing a line segment of points z sign-complementary to t. Clearly this line segment is $B(t)$. Under assumption 2.1 each nonempty $B(t)$ is a line segment in $A(t)$ having two endpoints. We now want to show that an endpoint of a line segment $B(t)$ is either z^0, or a solution point, or an endpoint of a line segment $B(t')$ with t' differing from t in just one component. An endpoint is characterized by the fact that n+1 variables are nonbasic. More precisely, at an endpoint z of $B(t)$ either λ is equal to 0 or 1 and for all j either z_j or s_j is nonbasic, or λ is basic and for exactly one index h both z_h and s_h are nonbasic. If λ is equal to 0 then $z=z^0$ and z is an endpoint of the unique line segment $B(t^0)$ with $t^0=\text{sgn}(Mz^0+q)$.

In the following lemmas we consider the endpoints of line segments in case λ is not equal to 0.

<u>Lemma 2.1</u>. Let z be an endpoint of a line segment $B(t)$. If λ is equal to 1, then z is a solution point.

<u>Proof</u>. Since $z\in A(t)$ and $\lambda=1$ we have that

$$z_j = b_j \qquad \text{if } t_j = +1$$

$$z_j = a_j \qquad \text{if } t_j = -1$$

$$\text{and} \qquad a_j \leq z_j \leq b_j \qquad \text{if } t_j = 0$$

Moreover, $t=\text{sgn}(Mz+q)$ since z also lies in $C(t)$ and because of assumption 2.1. Hence z is a solution point.

□

In the next lemma, let Z^b (Z^a) be the set of indices j for which $z_j^0=b_j$ ($z_j^0=a_j$).

Lemma 2.2. Let z be an endpoint of a line segment B(t) and let s=Mz+q. If at the point z, s_h becomes nonbasic, then z is a solution point if $I^+(t)\setminus\{h\}\subset Z^b$ and $I^-(t)\setminus\{h\}\subset Z^a$.

Proof. The conditions of the lemma imply that $z_j^0=b_j$ for all $j\in I^+(t)$ and $z_j^0=a_j$ for all $j\in I^-(t)$, $j\neq h$. Furthermore $t_j=\text{sgn } s_j$ for all $j\neq h$ and $s_h=0$. Therefore $j\in I^+(t)$ implies $s_j>0$ and $z_j=\lambda(b_j-z_j^0)+z_j^0=b_j$, and $j\in I^-(t)$ implies $s_j<0$ and $z_j=\lambda(a_j-z_j^0)+z_j^0=a_j$. Moreover, for all other indices j we have $s_j=0$ and $a_j\leq z_j\leq b_j$.

□

Lemma 2.3. Let z be an endpoint of a line segment B(t) and let s=Mz+q. If at z, s_h becomes nonbasic and $I^+(t)\setminus\{h\}$ contains at least one index j, $j\neq h$, not in Z^b or $I^-(t)\setminus\{h\}$ contains at least one index j, $j\neq h$, not in Z^a, then z is also an endpoint of B(t') with $t_h'=0$ and $t_j'=t_j$ for all $j\neq h$.

Proof. The conditions of the lemma imply that there is an index j, $j\neq h$, with $t_j'=+1$ and $z_j^0<b_j$ or an index k, $k\neq h$, with $t_k'=-1$ and $z_k^0>a_k$. Hence A(t') is not empty. Moreover, since $t_j'=t_j$, $j\neq h$, $t_h'=0$ and $t_h\in\{-1,1\}$, A(t') contains A(t) as a boundary facet and therefore z is in the boundary of A(t'). Finally, since $t_j=\text{sgn } s_j$, $j\neq h$, and $s_h=0$ we have that t'=sgn s. Consequently z lies in C(t') and is an endpoint of B(t').

□

Lemma 2.4. Let z be an endpoint of a line segment B(t) and let s=Mz+q. If at z, z_h becomes nonbasic then z is also an endpoint of B(t') with $t_j'=t_j$, $j\neq h$ and t_h' equal to either +1 or -1.

Proof. At z the variable z_h becomes nonbasic, i.e. for the λ defined in (2.2) holds

$$\lambda(a_h-z_h^0)\leq z_h-z_h^0 \leq\lambda(b_h-z_h^0)$$

with just one equality. If $z_h = z_h^0 + \lambda(a_h - z_h^0)$ then $z \in A(t')$ with $t'_h = -1$ and $t'_j = t_j$, $j \neq h$. On the other hand, if $z_h = z_h^0 + \lambda(b_h - z_h^0)$ then z lies in $A(t')$ with $t'_h = +1$ and $t'_j = t_j$ for all $j \neq h$. Finally, since $t = \text{sgn } s$ while $s_h = 0$ we have that $z \in Cl\{\bar{z} \mid \text{sgn } \bar{s} = t'\} = C(t')$. Hence z is an endpoint of $B(t')$.

□

The lemma's 2.3 and 2.4 say that if $\bar{z}$ is an endpoint of the line segment $B(t)$ and $\bar{z}$ is not a solution point, then $\bar{z}$ is an endpoint of the line segment $B(t')$. The nondegeneracy assumption guarantees that at $\bar{z}$ just one basic variable becomes nonbasic. This implies that t' is uniquely determined. So, linking the line segments $B(t)$ for various $t \in T$ together, the set $B = \bigcup_{t \in T} B(t)$ contains a piecewise linear path having the starting point z^0 as an endpoint. Since T consists of a finite number of elements t and since each $B(t)$ is either empty or a single line segment, the path in B originating at z^0 consists of a finite number of linear pieces and ends at a solution point $\bar{z}$. This path is generated by the algorithm and can be followed by a sequence of linear programming steps in a system of n linear equations.

3. Performance of the algorithm.

We consider now a point z on the path traced by the algorithm. For such a point we have that $z \in A(t) \cap C(t)$ for some sign vector $t \in T$. So, with $t = \text{sgn } s = \text{sgn } Mz+q$ we have that

$$z_j = \lambda(b_j - z_j^0) + z_j^0 \qquad \text{if } t_j = +1$$

$$z_j = \lambda(a_j - z_j^0) + z_j^0 \qquad \text{if } t_j = -1$$

$$\lambda(a_j - z_j^0) \le z_j - z_j^0 \le \lambda(b_j - z_j^0) \qquad \text{if } t_j = 0$$

with $0 \le \lambda \le 1$. Let $t' \in \{-1,1\}^n$ be a sign vector such that $t'_i = t_i$ if $t_i \neq 0$. Then we can rewrite z as

$$z = (1-\lambda)z^0 + \lambda v(t') - \Sigma_{h \in I^0(t)} \delta_h t'_h e(h) \tag{3.1}$$

for certain δ_h, $0\leq\delta_h\leq\lambda(b_h-a_h)$, where e(h) is the h-th unit vector, h=1,...,n. From (3.1) we obtain

$$s = Mz+q = (1-\lambda)Mz^0+\lambda Mv(t') - \sum_{h\in I^0(t)} \delta_h t'_h Me(h) + q. \qquad (3.2)$$

With $Mz^0+q=q^0$ and $Me(h)\equiv M_h$ this reduces to

$$s+\lambda M[z^0-v(t')] + \sum_{h\in I^0(t)} \delta_h t'_h M_h = q^0. \qquad (3.3)$$

Since t=sgn s, s is equal to $\sum_{h\notin I^0(t)} \mu_h t_h e(h)$ with $\mu_h=t_h s_h\geq 0$. All together we obtain that $z\in A(t)\cap C(t)$ if and only if

$$\lambda M[z^0-v(t')] + \sum_{h\in I^0(t)} \delta_h t'_h M_h + \sum_{h\notin I^0(t)} \mu_h t_h e(h) = q^0 \qquad (3.4)$$

for certain $0\leq\lambda\leq 1$, $0\leq\delta_h\leq\lambda(b_h-a_h)$ and $\mu_h\geq 0$. It will follow that t' is generated uniquely by the algorithm. The nondegeneracy assumption implies that any solution to the system (3.4) of n linear equations has at most one of the n+1 variables (λ,δ_h,μ_h) equal to its upper or lower bound. Hence, the linear path of points B(t), $t\in T$, can be followed by making a linear programming pivot step in the system (3.4). The performance of the algorithm to follow the piecewise linear path from z^0 to a solution of (2.1) can therefore be described in the next procedure, where z^0 is the initial point and $s^0=Mz^0+q$.

Step 0 (Initialisation). Set $t'=t=t^0=\text{sgn}(Mz^0+q)$. If $z^0=v(t^0)$ then (s^0,z^0) solves the problem. Otherwise, set $I^0(t)=I^0(t^0)=\emptyset$ and make a l.p. pivot step with the vector $M[z^0-v(t^0)]$ into the system

$$\sum_{h\notin I^0(t)} \mu_h t_h e(h) = Mz^0+q = q^0.$$

If μ_i becomes zero for some i, goto step 1. Otherwise the variable λ associated to $M[z^0-v(t^0)]$ becomes equal to 1 and the vertex $v(t^0)$ is a solution point (lemma 2.1).

<u>Step 1. (lemma 2.3.)</u>. Set $t_i=0$, $K^0(t)=I^0(t)\cup\{i\}$ and make a l.p. pivot step with $t_i^{\prime}M_i$ into the system

$$\lambda M[z^0-v(t')] + \Sigma_{h\in I^0(t)} \delta_h t_h^{\prime} M_h + \Sigma_{h\notin K^0(t)} \mu_h t_h e(h) = q^0.$$

If λ becomes equal to 1, a solution is found (lemma 2.1.). Otherwise goto step 2.

<u>Step 2</u>. Set $I^0(t)=K^0(t)$. If δ_i becomes 0 for some $i\in K^0(t)$ goto step 3. If δ_i becomes $\lambda(b_i-a_i)$ for some $i\in K^0(t)$, set $t_i^{\prime}$ equal to $-t_i^{\prime}$, adapt the column $M[z^0-v(t')]$ accordingly and make δ_i equal to 0 and goto step 3. If μ_i becomes 0 for some $i\notin K^0(t)$ and if for all other $h\notin K^0(t)$, $z_h^0=b_h$ if $t_h=+1$ and $z_h^0=a_h$ if $t_h=-1$, then a solution is found (lemma 2.2). Otherwise, return to step 1.

<u>Step 3. (lemma 2.4.)</u>. Set $t_i=t_i^{\prime}$, $K^0(t)=I^0(t)\setminus\{i\}$ and make a l.p. pivot step with $t_i e(i)$ in

$$\lambda M[z^0-v(t')] + \Sigma_{h\in K^0(t)} \delta_h t_h^{\prime} M_h + \Sigma_{h\notin I^0(t)} \mu_h t_h e(h) = q^0.$$

If λ becomes equal to 1, a solution is found (lemma 2.1). Otherwise goto step 2.

4. Some examples.

In this section we compare the 2n-ray or cubical algorithm initiated by Talman and Van der Heyden with the 2^n-ray or octahedral algorithm described in this paper. Remember that the cubical algorithm reduces to Lemkes algorithm when $z^0=0$ (and there are no upper bounds). If the solution point is on the interior of C^n both algorithms may solve the problem in n pivot steps (best cases). However, if a vertex of C^n is found as a solution point, the octahedral algorithm may solve the problem in only one step, while the cubical algorithm needs at least 2n-1 l.p. pivot steps. This difference is shown in the next example for n=2. In Talman and Van der Heyden a measure $t_0(z)$ is defined, called the "leading infeasibility". With some adaptions and for z^0 being an interior point, this measure is defined by

$t_0(z)=\max(\max\{s_j^+(z) \mid z_j<b_j\}, \max\{s_j^-(z) \mid z_j>a_j\})$. A point $\bar{z}$ is a solution if and only if $t_0(\bar{z})\leq 0$. Now consider figure 1 in which the sign structure of problem 1 is given. The problem has three solution points namely the vertex $v=v((-1,+1)^T)$ and the points x^* and x^1. The starting

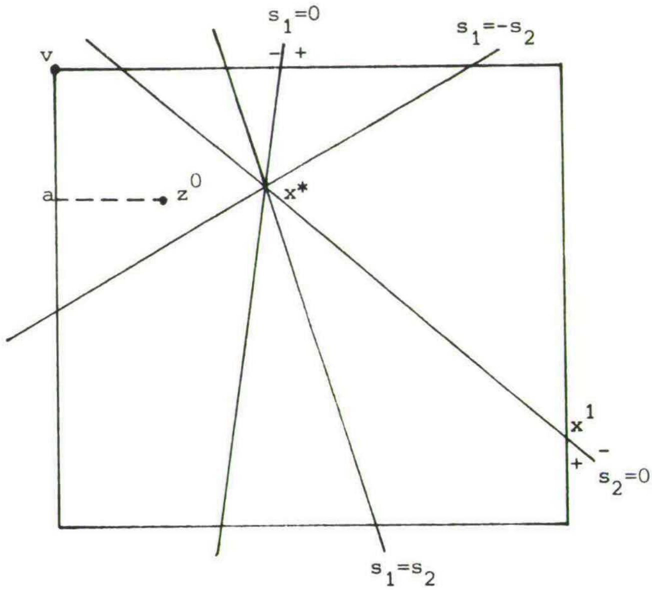

Figure 1.

point is z^0 in which $-s_1^0>s_2^0>0$ where $s^0=Mz^0+q$. Clearly $t_0(z^0)=-s_1^0$ and the cubical algorithm decreases z_1 by making a l.p. pivot step, until the point a is reached. For all points z on the line segment $[z^0,a)$, $t_0(z)=-s_1$. However, at the point a we have $t_0(a)=s_2(a)$, causing a discontinuity in t_0. To overcome this, an additional l.p. pivot step is made at a to decrease t_0 from $-s_1(a)$ to $s_2(a)$. After this z_2 is increased until the solution point v is reached, at which point t_0 is decreased from s_2 to 0 by making another l.p. pivot step. Since the last step is redundant, the algorithm needs $2n-1=3$ l.p. steps. On the other hand, the octahedral algorithm reaches the solution point v in just one l.p. step, because the sign pattern does not change on the line segment between z^0 and this vertex.

In figure 2 the solution points are x^*, x^1 and x^2. Now the cubical algorithm goes from z^0 to the point a, then makes a l.p. step to decrease t_0 from $-s_1(a)$ to $s_2(a)$ and finally goes from a to x^2, so that again 3 l.p. steps are needed to find a solution. The octahedral algorithm goes from z^0 to b and follows then the line $s_2=0$ until x^2 is

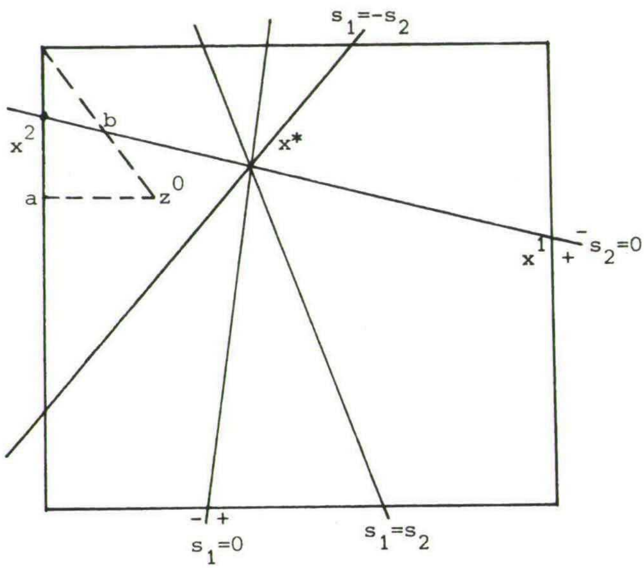

Figure 2.

reached, implying that 2 steps are needed. In figure 3 both algorithms find the unique solution point x^* in just two steps. At z^0 we have that $s_1^0>s_2^0>0$. The cubical algorithm follows the path $z^0 \to a \to x^*$, and the octahedral algorithm follows the path $z^0 \to b \to x^*$.

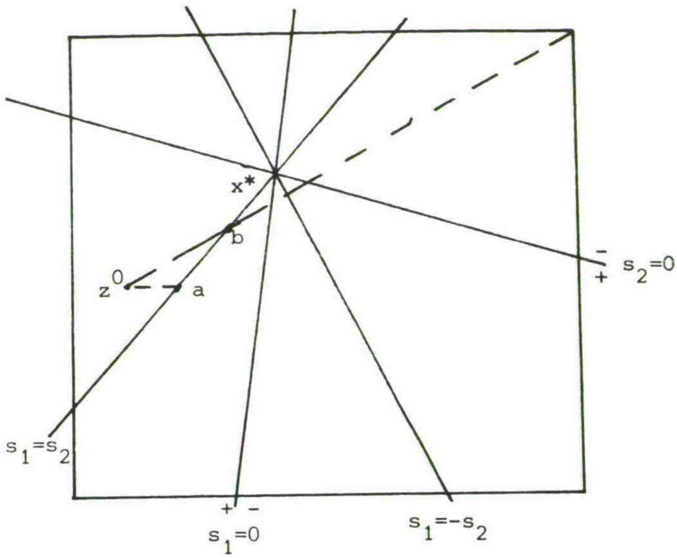

Figure 3.

In general, let k be the dimension of the face of C^n containing the solution found by the algorithm in its interior. Then the minimum number of steps (best cases) to find the solution point is 2n-k for the cubical algorithm (2n-1 if k=0) and k+1 for the octahedral algorithm (n if k=n). In particular if k is small compared to n, the best case for the cubical algorithm (or generalized Lemke's algorithm) is considerably worse than the best case for the octahedral algorithm. Of course, for both algorithms the minimum number decreases if the starting point is on the boundary of C^n. For instance, if in figure 2 z^0 is chosen to be a, both algorithms need only 1 step.

We now consider the worst cases for n=2. This is done in the figures 4 and 5. In figure 4 we show the maximum number of l.p. pivot steps to find a solution for the cubical algorithm and in figure 5 we show this number for the octahedral algorithm. The cubical algorithm traces the path z^0, a, b, c, d, e, f, g, v, the latter point being the unique solution point. So, the algorithm initially increases z, until a is reached. Then z_2 is increased until $s_1=s_2$. The latter line is followed as long as $z_1>z_1^0$. As soon as $z_1=z_1^0$, the algorithm continues by increasing z_2 until $z_2=b_2$, then z_1 is decreased until

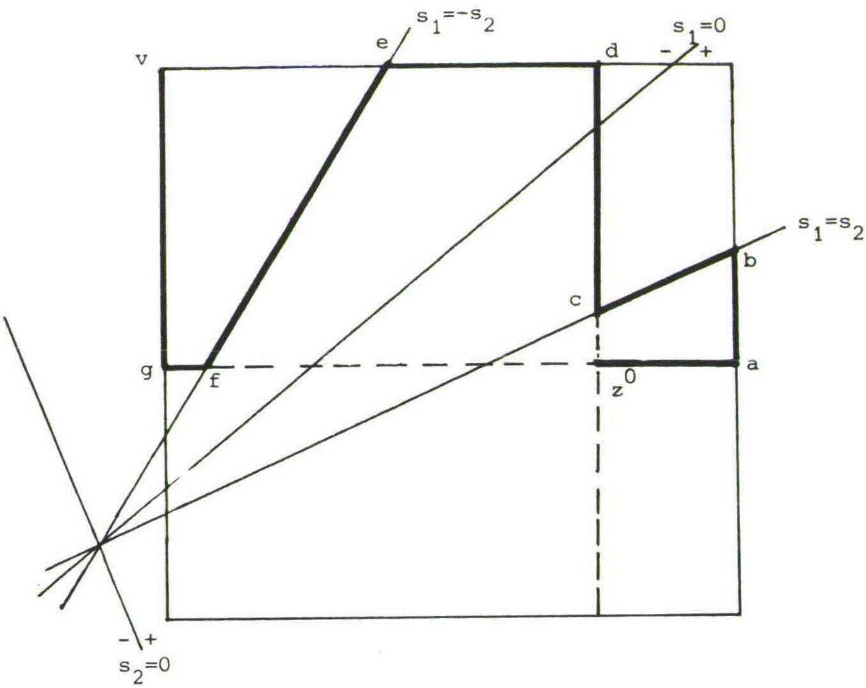

Figure 4: The worst case for the cubical algorithm, n=2.

$s_1=-s_2$. Finally this line is followed until $z_2=z_2^0$, then z_1 is decreased until $z_1=a_1$ and then z_2 is increased until the solution point v is reached. Including the additional l.p. steps to overcome the discontinuity of $t_0(z)$ at the points a, d and g, this takes 11 l.p. steps. We now consider the worst case for the octahedral algorithm. This is shown in figure 5. In this figure there are 3 solution points, namely x^*, x^1 and v. The algorithm traces the path z^0, a, b, c, d, v, which takes 5 l.p. steps. At z^0 we have that $s_1^0>0$ and $s_2^0<0$, implying a search in the direction $d((+1,-1)^T)$. Then $s_2=0$ is followed until the ray $A((+1,+1)^T)$ is reached. This ray is followed until c is found where $s_1=0$. The line $s_1=0$ is followed until point d and finally $A((-1,+1)^T)$ is followed until the solution point v is found. The examples above show the superiority of the octahedral algorithm above the cubical algorithm for n=2. It is our conjecture that the differences in the number of l.p. steps will increase dramatically if the dimension of the problem growths.

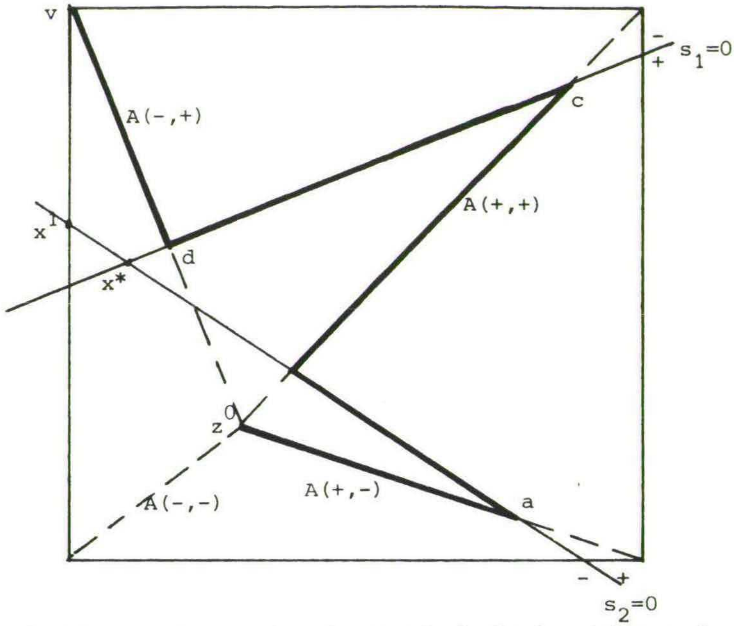

Figure 5: The worst case for the octahedral algorithm, n=2. A(+,-) denotes A(t) with t=(+1,-1) , etc.

The algorithm described in this paper can easily be adapted for the case that some of the a_i's or b_i's are not finite. For simplicity, let us consider the classical LCP where $a_i=0$ and $b_i=+\infty$ for all i. Then we redefine the sets A(t), $t\epsilon T$, by

$$A(t) = \{z\epsilon R^n_+ | z_j=z^0_j+\lambda \text{ if } t_j=+1,\ z^0_j\le z_j\le z^0_j+\lambda \text{ if } t_j=0, \text{ and } z_j=z^0_j \text{ if } t_j=-1, \lambda\ge 0\}$$

if $t_j=+1$ for at least one index j, by

$$A(t) = \{z\epsilon R^n_+ | z_j=(1-\lambda)z^0_j \text{ if } t_j=0, \text{ and } z_j=(1-\lambda)z^0_j \text{ if } t_j=-1, 0\le\lambda\le 1\}$$

if $t\le 0$ and $t_j=-1$ for at least one index j for which $z^0_j>0$, and $A(t)=\emptyset$ otherwise. Again let $B(t)=A(t)\cap C(t)$ with C(t) as before, then the set $B=\cup_t B(t)$ contains a piecewise linear path from z^0 which can be followed by subsequent linear programming pivot steps in a system of linear equations similar to (3.4). The piecewise linear path originating at z^0 leads within a finite number of steps either to a solution point or terminates with a half-line to infinity.

It can easily be shown that Evers'condition is sufficient for convergence of the algorithm if a solution exists (see Jones [3]). The case n=2 is illustrated in figure 6.

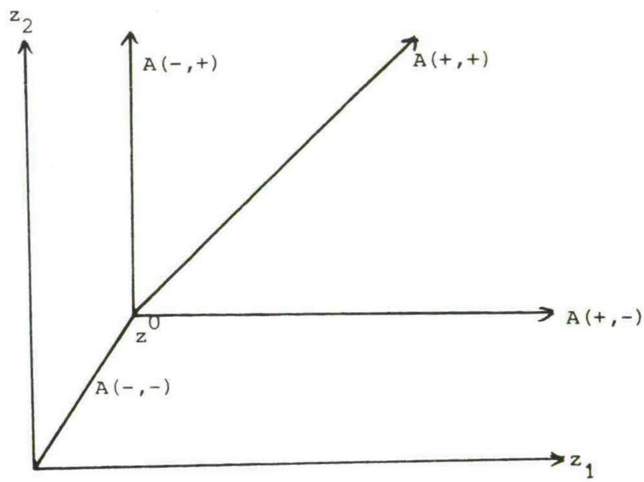

Figure 6: The sets A(t) if $C^n = R^n_+$, n=2.

In case $z^0=0$ the algorithm differs from Lemke's original algorithm. The new algorithm leaves $z^0=0$ by increasing all the z_j's for which q_j is positive whereas Lemke's algorithm increases only the z_i for which q_i is maximal. The latter algorithm therefore can leave the starting point $z^0=0$ in n directions and the algorithm described in this paper in 2^n-1 directions. In the case that $z^0=0$ and all the b_i's are plus infinite the worst case of the cubical algorithm needs one step less than the worst case of the octahedral algorithm (n=2). In all other cases the new algorithm performs better.

References.

[1] I.D. Everts, "Het lineaire complementariteitsprobleem", Landbouw-Economisch Instituut, Den Haag, The Netherlands (1982).

[2] G.B. Garcia and F.J. Gould, "Studies in linear complementarity", Center for Mathematical Studies in Business and Economics, University of Chicago, Chicago, Ill. (1980).

[3] P.C. Jones, "A note on the Talman, Van der Heyden linear complementarity algorithm", Mathematical Programming 25 (1983), 122-124.

[4] N. Josephy, "Newton's method for generalized equations", Technical Summary Report 1965, Mathematics Research Center, University of Wisconsin, Madison, Wi. (1979).

[5] G. van der Laan and A.J.J. Talman, "A class of simplicial restart fixed points algorithms without an extra dimension", Mathematical Programming 20 (1981), 33-48.

[6] C.E. Lemke, "Bimatrix equilibrium points and mathematical programming", Management Science 11 (1965), 681-689.

[7] L. Mathiesen and T. Hansen, "An equilibrium model for an open economy with institutional constraints on factor prices", in: W. Forster, ed., Numerical Solution of Highly Nonlinear Problems (North-Holland, Amsterdam, 1980) pp. 337-360.

[8] P.M. Reiser, "A modified integer labelling for complementarity algorithms", Mathematics of Operations Research 6 (1981), 129-139.

[9] A.J.J. Talman and L. Van der Heyden, "Algorithms for the linear complementarity problem which allow an arbitrary starting point", in: B.C. Eaves et all.,eds., Homotopy Methods and Global Convergence (Plenum Press, New York, 1983), pp. 267-285.

[10] A.H. Wright, "The octahedral algorithm, a new simplicial fixed point algorithm", Mathematical Programming 21 (1981), 47-69.

IN 1984 REEDS VERSCHENEN

138 G.J. Cuypers, J.P.C. Kleijnen en J.W.M. van Rooyen
Testing the Mean of an Asymetric Population:
Four Procedures Evaluated

139 T. Wansbeek en A. Kapteyn
Estimation in a linear model with serially correlated errors when observations are missing

140 A. Kapteyn, S. van de Geer, H. van de Stadt, T. Wansbeek
Interdependent preferences: an econometric analysis

141 W.J.H. van Groenendaal
Discrete and continuous univariate modelling

142 J.P.C. Kleijnen, P. Cremers, F. van Belle
The power of weighted and ordinary least squares with estimated unequal variances in experimental design

143 J.P.C. Kleijnen
Superefficient estimation of power functions in simulation experiments

144 P.A. Bekker, D.S.G. Pollock
Identification of linear stochastic models with covariance restrictions.

145 Max D. Merbis, Aart J. de Zeeuw
From structural form to state-space form

146 T.M. Doup and A.J.J. Talman
A new variable dimension simplicial algorithm to find equilibria on the product space of unit simplices.

147 G. van der Laan, A.J.J. Talman and L. Van der Heyden
Variable dimension algorithms for unproper labellings.

148 G.J.C.Th. van Schijndel
Dynamic firm behaviour and financial leverage clienteles

149 M. Plattel, J. Peil
The ethico-political and theoretical reconstruction of contemporary economic doctrines

150 F.J.A.M. Hoes, C.W. Vroom
Japanese Business Policy: The Cash Flow Triangle
an exercise in sociological demystification

151 T.M. Doup, G. van der Laan and A.J.J. Talman
The $(2^{n+1}-2)$-ray algorithm: a new simplicial algorithm to compute economic equilibria

IN 1984 REEDS VERSCHENEN (vervolg)

152 A.L. Hempenius, P.G.H. Mulder
Total Mortality Analysis of the Rotterdam Sample of the Kaunas-Rotterdam Intervention Study (KRIS)

153 A. Kapteyn, P. Kooreman
A disaggregated analysis of the allocation of time within the household.

154 T. Wansbeek, A. Kapteyn
Statistically and Computationally Efficient Estimation of the Gravity Model.

155 P.F.P.M. Nederstigt
Over de kosten per ziekenhuisopname en levensduurmodellen

156 B.R. Meijboom
An input-output like corporate model including multiple technologies and make-or-buy decisions

157 P. Kooreman, A. Kapteyn
Estimation of Rationed and Unrationed Household Labor Supply Functions Using Flexible Functional Forms

158 R. Heuts, J. van Lieshout
An implementation of an inventory model with stochastic lead time

159 P.A. Bekker
Comment on: Identification in the Linear Errors in Variables Model

160 P. Meys
Functies en vormen van de burgerlijke staat
Over parlementarisme, corporatisme en autoritair etatisme

161 J.P.C. Kleijnen, H.M.M.T. Denis, R.M.G. Kerckhoffs
Efficient estimation of power functions

162 H.L. Theuns
The emergence of research on third world tourism: 1945 to 1970; An introductory essay cum bibliography

163 F. Boekema, L. Verhoef
De "Grijze" sector zwart op wit
Werklozenprojecten en ondersteunende instanties in Nederland in kaart gebracht

164 G. van der Laan, A.J.J. Talman, L. Van der Heyden
Shortest paths for simplicial algorithms

165 J.H.F. Schilderinck
Interregional structure of the European Community
Part II: Interregional input-output tables of the European Community 1959, 1965, 1970 and 1975.

IN (1984) REEDS VERSCHENEN (vervolg)

166 P.J.F.G. Meulendijks
An exercise in welfare economics (I)

167 L. Elsner, M.H.C. Paardekooper
On measures of nonnormality of matrices.

IN 1985 REEDS VERSCHENEN

168 T.M. Doup, A.J.J. Talman
A continuous deformation algorithm on the product space of unit simplices

169 P.A. Bekker
A note on the identification of restricted factor loading matrices

170 J.H.M. Donders, A.M. van Nunen
Economische politiek in een twee-sectoren-model

171 L.H.M. Bosch, W.A.M. de Lange
Shift work in health care

172 B.B. van der Genugten
Asymptotic Normality of Least Squares Estimators in Autoregressive Linear Regression Models

173 R.J. de Groof
Geïsoleerde versus gecoördineerde economische politiek in een twee-regiomodel

174 G. van der Laan, A.J.J. Talman
Adjustment processes for finding economic equilibria

175 B.R. Meijboom
Horizontal mixed decomposition

176 F. van der Ploeg, A.J. de Zeeuw
Non-cooperative strategies for dynamic policy games and the problem of time inconsistency: a comment

177 B.R. Meijboom
A two-level planning procedure with respect to make-or-buy decisions, including cost allocations

178 N.J. de Beer
Voorspelprestaties van het Centraal Planbureau in de periode 1953 t/m 1980

178a N.J. de Beer
BIJLAGEN bij Voorspelprestaties van het Centraal Planbureau in de periode 1953 t/m 1980

179 R.J.M. Alessie, A. Kapteyn, W.H.J. de Freytas
De invloed van demografische factoren en inkomen op consumptieve uitgaven

180 P. Kooreman, A. Kapteyn
Estimation of a game theoretic model of household labor supply

181 A.J. de Zeeuw, A.C. Meijdam
On Expectations, Information and Dynamic Game Equilibria

182 Cristina Pennavaja
Periodization approaches of capitalist development.
A critical survey

183 J.P.C. Kleijnen, G.L.J. Kloppenburg and F.L. Meeuwsen
Testing the mean of an asymmetric population: Johnson's modified T test revisited

184 M.O. Nijkamp, A.M. van Nunen
Freia versus Vintaf, een analyse

185 A.H.M. Gerards
Homomorphisms of graphs to odd cycles

186 P. Bekker, A. Kapteyn, T. Wansbeek
Consistent sets of estimates for regressions with correlated or uncorrelated measurement errors in arbitrary subsets of all variables

187 P. Bekker, J. de Leeuw
The rank of reduced dispersion matrices

188 A.J. de Zeeuw, F. van der Ploeg
Consistency of conjectures and reactions: a critique

189 E.N. Kertzman
Belastingstructuur en privatisering

190 J.P.C. Kleijnen
Simulation with too many factors: review of random and group-screening designs

191 J.P.C. Kleijnen
A Scenario for Sequential Experimentation

192 A. Dortmans
De loonvergelijking
Afwenteling van collectieve lasten door loontrekkers?

193 R. Heuts, J. van Lieshout, K. Baken
The quality of some approximation formulas in a continuous review inventory model

194 J.P.C. Kleijnen
Analyzing simulation experiments with common random numbers

195 P.M. Kort
Optimal dynamic investment policy under financial restrictions and adjustment costs

196 A.H. van den Elzen, G. van der Laan, A.J.J. Talman
Adjustment processes for finding equilibria on the simplotope

197 J.P.C. Kleijnen
Variance heterogeneity in experimental design

198 J.P.C. Kleijnen
Selecting random number seeds in practice

199 J.P.C. Kleijnen
Regression analysis of simulation experiments: functional software specification

Bibliotheek K. U. Brabant
17 000 01059741 8