Simplicial approximation of solutions to the nonlinear complementarity problem with lower and upper bounds

Talman, A.J.J.; van der Laan, G.

Published in:
Mathematical Programming

Publication date:
1987

Mathematical Programming 38 (1987) 1–15
North-Holland

SIMPLICIAL APPROXIMATION OF SOLUTIONS TO THE NONLINEAR COMPLEMENTARITY PROBLEM WITH LOWER AND UPPER BOUNDS

G. van der LAAN
Department of Economics and Econometrics, Free University, Amsterdam, The Netherlands

A. J. J. TALMAN
Department of Econometrics, Tilburg University, Tilburg, The Netherlands

Received 27 December 1983
Revised manuscript received 13 November 1986

Ideas of a simplicial variable dimension restart algorithm to approximate zero points on R^n developed by the authors and of a linear complementarity problem pivoting algorithm are combined to an algorithm for solving the nonlinear complementarity problem with lower and upper bounds. The algorithm can be considered as a modification of the $2n$-ray zero point finding algorithm on R^n. It appears that for the new algorithm the number of linear programming pivot steps is typically less than for the $2n$-ray algorithm applied to an equivalent zero point problem. This is caused by the fact that the algorithm utilizes the complementarity conditions on the variables.

Key words: Simplicial algorithm, triangulation, nonlinear complementarity problem.

1. Introduction

For a mapping f from the n-dimensional Euclidean space R^n into itself, the nonlinear complementarity problem (NLCP) is to find a vector x in R^n such that

$$x \geq 0, \quad f(x) \geq 0 \quad \text{and} \quad x^{\mathrm{T}} f(x) = 0,$$

i.e., x and $f(x)$ are orthogonal and have nonnegative components. The NLCP can be solved by methods for finding a zero point in several ways. Converting the NLCP into a zero finding problem, Merrill [16] and several other authors (see e.g. [1]) utilized simplicial fixed point algorithms to find an approximate solution. Using a reformulation of the NLCP due to Mangasarian [15], in which the zero finding problem can be made as smooth as desired, Watson [23] applied the homotopy or continuation method of Chow, Mallet-Paret and Yorke [2] to solve the problem. Instead of reformulating the NLCP as a zero finding problem, other authors adjusted simplicial fixed point algorithms to solve the NLCP directly, see e.g. [4], [5], [7] or [14]. This approach will be followed in this paper.

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

Simplicial methods are based on a subdivision of R^n into n-dimensional simplices and on a function assigning to each vertex of the subdivision a label. Then the algorithm searches for a so-called completely labelled simplex. Such a simplex yields an approximate solution to the problem. A special subclass of algorithms of this type are the variable dimension algorithms initiated by van der Laan and Talman [11]. These algorithms generate a path of simplices of varying dimension starting with a zero-dimensional simplex being a grid point of the subdivision, the starting point, and terminating in a completely labelled simplex yielding an approximate solution x^*. In a finer subdivision a restart can be made with x^* as the new starting point. The various algorithms of this type differ in the number of rays along which the arbitrarily chosen starting point v can be left. Until now, algorithms of this type have been developed with $n+1$ rays [11], $2n$ rays [12], 2^n rays [24], 3^n-1 rays [10], and with 2 rays [18, 25].

The $2n$-ray algorithm has been adapted in Talman and Van der Heyden [20] to solve the linear complementarity problem. In this paper we will modify the $2n$-ray algorithm in a similar way such that it can be applied to solve the NLCP directly. Therefore we utilize explicitly the complementarity conditions on the variables. Doing so the number of linear programming steps will be typically less than for the original $2n$-ray algorithm applied to the equivalent zero finding problem.

It should be observed that our approach differs completely from another "variable dimension" algorithm to solve the NLCP, namely the direct algorithm of Habetler and Kostreva [6]. This algorithm moves among subsets of $\{1, \ldots, n\}$ until a subset I^* is found which leads to a complementarity point s^* in R^n_+. Such a point yields a solution point x^*. For each subset I in the sequence, a zero of the function f^I defined by $f^I_i(x) = f_i(x)$ if $i \in I$ and $f^I_i(x) = x_i$ if $i \notin I$ has to be approximated. Since for a zero x' of f^I holds that $x'_i = 0$ if $i \notin I$, it follows that then an $|I|$-dimensional problem has to be solved with $|I|$ the cardinality of I. So, a *sequence* of problems of varying dimensions has to be solved. Each zero point problem in this sequence can be solved by a simplicial algorithm.

This paper is organized as follows. In Section 2 the steps of the $2n$-ray algorithm for the zero point problem on R^n are given and we discuss how the (generalized) nonlinear complementarity problem can be solved by converting the problem in a zero-point problem on R^n. In Section 3 the piecewise linear path of the modified $2n$-ray algorithm is derived in order to solve the NLCP directly, whereas Section 4 gives the steps of the algorithm in terms of a path of simplices generating procedure. Some concluding remarks are made in Section 5.

2. The $2n$-ray algorithm and the generalized NLCP

In this section we give a short description of the $2n$-ray algorithm to solve the zero point problem $f(x) = 0$ with f a continuous function from R^n into itself. For a more detailed description we refer to [12] and [19]. In the sequel, let I_n be the set of integers $\{1,\dots,n\}$ and let K_n be the set $\{-n,-n+1,\dots,-1,1,\dots,n\}$. The h-th unit vector in R^n (R^{n+1}) will be denoted by $e(h)$ ($e'(h)$), and for $h\in I_n$ we define $e(-h)=-e(h)$. Furthermore, let Z be the collection of subsets of K_n such that for each $T\in Z$ not both j and $-j$ belong to T, $j=1,\dots,n$. For $T\in Z$, $s(T)\in R^n$ is the sign vector with $s_j(T)=1$ when $j\in T$, $s_j(T)=-1$ when $-j\in T$, and with $s_j(T)=0$ otherwise. Finally, t or $|T|$ denotes the cardinality of T and $\underline{T}=\{j\in I_n\,|\,\text{neither } j \text{ nor } -j \text{ is in } T\}$, $T\in Z$.

Definition 2.1. Given some $v\in R^n$, for each $T\in Z$, $A(T)$ is the subset of R^n such that

(i) $x_j=v_j$ when $j\in\underline{T}$,
(ii) $x_j\geqslant v_j$ when $j\in T$,
(iii) $x_j\leqslant v_j$ when $-j\in T$.

Observe that $A(T)$ is a t-dimensional subset of R^n. When $T=\emptyset$, $A(T)=\{v\}$, where v is the point where the algorithm will start. Furthermore, for all $T\in Z$, we have bd $A(T)=\bigcup_{j\in T}A(T\backslash\{j\})$. Now, let Γ be a triangulation of R^n such that each subset $A(T)$ is triangulated by Γ into t-simplices. Such a triangulation is e.g. the K'-triangulation, proposed by Todd [21] and used in [12]. For $T\in Z$ we call a t-simplex $\sigma(w^1,\dots,w^{t+1})$ in $A(T)$, with vertices $w^1,\dots,w^{t+1}$, T-complete if the $(n+1)\times(n+2)$-system of linear equations (see e.g. Todd [22])

$$\sum_h \lambda_h f'(w^h)+\sum_{i\in\underline{T}}\mu_i e'(i)+\beta s'(T)=e'(n+1) \tag{2.1}$$

where $f'(x)=(f(x)^{\mathrm T},1)^{\mathrm T}$ and $s'(T)=(s(T)^{\mathrm T},1)^{\mathrm T}$, has a solution $\lambda_h^*\geqslant 0$, $h=1,\dots,t+1$, $\beta^*\geqslant 0$, and $-\beta^*\leqslant\mu_i^*\leqslant\beta^*$, $i\in\underline{T}$.

Assuming nondegeneracy, the system (2.1) has two basic solutions, if any, i.e., two solutions with exactly one of the constraints binding. The whole line segment between these two solutions forms the set of solutions to (2.1). Except when $T=\emptyset$ or $\beta^*=0$, each basic solution to (2.1) is also a basic solution with respect to exactly one other simplex adjacent to σ. This new simplex is uniquely determined by the binding constraint. When, at a basic solution, $\beta^*=0$, then the point $\sum_i\lambda_i^*w^i$ is an approximate zero point of f and the algorithm terminates. The $2n$-ray algorithm generates the sequence of adjacent simplices with T-complete common facets in $A(T)$, $T\in Z$, which starts for $T=\emptyset$ in v and terminates with a simplex where $\beta^*=0$ at a basic solution.

In fact the line segment of solutions to (2.1) determines a piecewise linear path of points $x=\sum_i\lambda_iw^i/\sum_i\lambda_i$ in σ with the following properties. Let F be the piecewise linear approximation of f induced by the triangulation Γ, i.e. $F(x)=\sum_i\lambda_if(w^i)$ when $x=\sum_i\lambda_iw^i$ with $\sum_i\lambda_i=1$ and $\lambda_i\geqslant 0$, $i=1,\dots,t+1$, and where $\sigma(w^1,\dots,w^{t+1})$ is a t-simplex in Γ containing x. Then, according to (2.1), for certain $\alpha\geqslant 0$

$$\begin{aligned} &|F_j(x)| \leq \alpha \quad \text{and } x_j = v_j \quad \text{when } j \in \underline{T}, \\ &F_j(x) = -\alpha \quad \text{and } x_j \geq v_j \quad \text{when } j \in T, \\ &F_j(x) = \alpha \quad \text{and } x_j \leq v_j \quad \text{when } -j \in T. \end{aligned} \tag{2.2}$$

When $T \neq \emptyset$, α is equal to $\alpha(x) = \max_j |F_j(x)|$. For varying $T \in Z$, starting with $T = \emptyset$ at v, the algorithm therefore generates a piecewise linear path of points satisfying (2.2). The path either goes to infinity or terminates at an x^* where $F(x^*) = 0$.

In this paper we will modify the $2n$-ray algorithm in order to solve the following problem which is a slight generalization of the original NLCP. For given vectors a and b in R^n with $a_i < b_i$ for all i, find x, $a \leq x \leq b$, such that for all $i = 1, \ldots, n$,

$$\begin{aligned} &x_i = a_i \quad \text{implies } f_i(x) \geq 0, \\ &a_i < x_i < b_i \quad \text{implies } f_i(x) = 0, \\ &x_i = b_i \quad \text{implies } f_i(x) \leq 0, \end{aligned} \tag{2.3}$$

where f is a continuous function from the set $C = \{x \in R^n \mid a \leq x \leq b\}$ into R^n. Such problems arise in game theory, economic modelling and constrained optimization. We allow components of a to be minus infinity and components of b to be plus infinity. When $a = 0$ and all components of b are plus infinity, we have the classical NLCP on R^n_+.

First we will discuss how the above described $2n$-ray algorithm on R^n can be used in order to solve problem (2.3) by converting the latter into a zero finding problem. For $z \in R^n$, let $p(z)$ be the projection on C defined by

$$\begin{aligned} &p_i(z) = a_i \quad \text{if } z_i < a_i, \\ &p_i(z) = z_i \quad \text{if } a_i \leq z_i \leq b_i, \\ &p_i(z) = b_i \quad \text{if } z_i > b_i. \end{aligned}$$

Then, x in C is a solution to problem (2.3) if and only if there is a z in R^n which satisfies both $x = p(z)$ and the system of equations (see e.g. Kojima and Saigal [8])

$$g(z) = f(p(z)) - p(z) + z = 0. \tag{2.4}$$

So, to find a solution to problem (2.3) we could apply the $2n$-ray algorithm on R^n to problem (2.4). Notice that the system (2.4) has a special structure in the sense that the left hand side $g(z)$ is partial linear on each piece of R^n which consists of the points which are projected onto a common face of C. More precisely, g is linear in the variables z_j for those indices j with $z_j \notin [a_j, b_j]$ given the other variables. This allows for combining simplices outside C to polyhedra. In this way the partial linear structure can be exploited in the $2n$-ray algorithm in order to save linear programming steps when tracing the piecewise linear path from v to an approximate solution z^* of $g(z) = 0$. The projection $p(z^*)$ is then an approximate solution to problem (2.3). Assuming nondegeneracy the point z^* always lies in the interior of a polyhedron of the underlying subdivision, although the projection $p(z^*)$ may lie on a lower-dimensional face of C. In the latter case, in order to improve the accuracy of the approximation the algorithm has to be restarted outside C and needs as in the case that z^* lies in C at least $n+1$ linear programming and replacement steps to find a new approximate solution to (2.4).

In the next section we propose a modification of the $2n$-ray algorithm on C which exploits explicitly the complementarity structure on the variables. In fact the modified algorithm will follow the (piecewise linear) projection on C of the piecewise linear path generated by the $2n$-ray algorithm applied to (2.4). In particular, if the $2n$-ray method on R^n restarts outside C at say z^*, the modified algorithm on C starts at the point $x^* = p(z^*)$ in a lower dimensional face of C. If k is the dimension of this face, then the minimum number of linear programming steps reduces from $n+1$ to $k+1$. So, the $2n$-ray algorithm applied to (2.4) typically needs more l.p. pivot steps than the algorithm on C, especially when the dimension of the face of C on which the latter method is (re)started is rather low.

3. The piecewise linear path

In this section we derive the piecewise linear path of the algorithm on C from the path followed by the $2n$-ray algorithm applied to system (2.4). Therefore we need a triangulation of R^n satisfying the following properties:

(a) the restriction of the triangulation to C is a triangulation of C itself;

(b) for any t-dimensional simplex τ with vertices $y^1, \ldots, y^{t+1}$ not in C the projection of τ on C is a k-simplex σ $(k<t)$ in the boundary of C such that the set of projections of the vertices of τ coincides with the set of vertices of σ.

Such a triangulation is obtained from the K'-triangulation by choosing an appropriate grid size. In the sequel we restrict ourselves to this triangulation. Therefore, let $m_1, m_2, \ldots, m_n$ be n positive integers and let d_i be defined by $d_i = (b_i - a_i)/m_i$, $i = 1, \ldots, n$. Furthermore, let D be the $n \times n$ diagonal matrix with the jth diagonal element equal to d_j, $j = 1, \ldots, n$. The set of grid points of the K'-triangulation is the set

$$\left\{x \in R^n \,\middle|\, x = a + \sum_i k_i De(i),\ k_i \text{ is the integer for all } i\right\}.$$

Then for an arbitrarily chosen grid point v in R^n the set $A(T)$, $T \in Z$, is triangulated by the collection of t-dimensional simplices $(t = |T|)$ $\sigma(y^1, \gamma(T))$ with vertices $y^1, \ldots, y^{t+1}$ such that

(i) y^1 is a grid point in $A(T)$

(ii) $\gamma(T) = (\gamma_1, \ldots, \gamma_t)$ is a permutation of the elements of T

(iii) $y^{i+1} = y^i + De(\gamma_i)$, $i = 1, \ldots, t$.

Now, let G be the piecewise linear approximation to g defined in (2.4) with respect to the triangulation of R^n and let F be the piecewise linear approximation to the function f. Since $g(z) = f(p(z)) - p(z) + z$ and the triangulation satisfies property (b) it follows that

$$G(z) = F(p(z)) - p(z) + z. \tag{3.1}$$

Furthermore, let the grid point v be the starting point of the $2n$-ray algorithm to find a zero point of G. As shown in Section 2 the algorithm traces a piecewise linear path from v to a solution point. Suppose that z is a point on this path. From (2.2) we know that for certain $\alpha > 0$ the following holds:

Condition N

$$|G_j(z)| \leq \alpha \quad \text{if } z_j = v_j,$$
$$G_j(z) = -\alpha \quad \text{if } z_j > v_j,$$
$$G_j(z) = \alpha \quad \text{if } z_j < v_j.$$

Now consider the projection $x = p(z)$ on C and suppose for simplicity that $v \in C$. If $a_j < x_j < b_j$ then $x_j = z_j$ and hence $F_j(x) = G_j(z)$. However, if $x_j = b_j$ then $z_j \geq x_j$ and hence $F_j(x) = G_j(z) + x_j - z_j \leq G_j(z)$. If $v_j < b_j$ then $G_j(z) = -\alpha$ and hence $F_j(x) \leq -\alpha$. If $v_j = b_j$ (and hence $> a_j$) then either $z_j = b_j$ and $|F_j(x)| = |G_j(z)| \leq \alpha$ or $z_j > b_j$ and $F_j(x) \leq G_j(z) = -\alpha$. Analogously it follows for the case that $x_j = a_j \geq z_j$ that $F_j(x) \geq \alpha$ if $a_j < v_j$ and $F_j(x) \geq -\alpha$ if $a_j = v_j$. Combining these cases we obtain from condition N that at $x = p(z)$ for some $\alpha > 0$ the following holds:

Condition P

$$F_j(x) = -\alpha \quad \text{if } v_j < x_j < b_j,$$
$$F_j(x) \leq \alpha \quad \text{if } x_j = v_j > a_j,$$
$$F_j(x) \leq -\alpha \quad \text{if } x_j = b_j > v_j,$$
$$F_j(x) = \alpha \quad \text{if } a_j < x_j < v_j,$$
$$F_j(x) \geq -\alpha \quad \text{if } x_j = v_j < b_j,$$
$$F_j(x) \geq \alpha \quad \text{if } x_j = a_j < v_j.$$

Observe that $|F_j(x)| \leq \alpha$ if $a_j < x_j = v_j < b_j$.

In the next section we will derive an algorithm to follow the piecewise linear path of points x in C which satisfy condition P and which is the projection on C of the piecewise linear path followed by the $2n$-ray algorithm. As soon as α becomes equal to zero in a point x^* we obtain that

$$F_j(x^*) = 0 \quad \text{if } a_j < x_j^* < b_j,$$
$$F_j(x^*) \geq 0 \quad \text{if } x_j^* = a_j,$$
$$F_j(x^*) \leq 0 \quad \text{if } x_j^* = b_j.$$

So, such a point x^* is a solution to the NLCP with respect to the piecewise linear approximation F to f and is an approximate solution to problem (2.3). It is well known that this approximation will be more accurate when the mesh of the triangulation becomes smaller.

4. The path following algorithm

The piecewise linear path of points from v satisfying condition P is traced by a sequence of adjacent simplices of varying dimension in the triangulation K' of R^n restricted to C. Each such simplex is the projection on C of one or a sequence of simplices generated by the $2n$-ray algorithm. Let $A'(T')$ be the restriction of $A(T')$ to C and let x be a point in $A(T')$, $T' \in Z$. Then the projection $p(x)$ on C lies in the face $A'(T, U)$ of the set $A'(T \cup U)$ defined by

$$A'(T, U) = \{x \in A'(T \cup U) \mid x_j = a_j \text{ if } -j \in U \text{ and } x_j = b_j \text{ if } j \in U\},$$

with T and U given by

$$T = \{j \in T' \mid x_j < b_j\} \cup \{-j \in T' \mid x_j > a_j\}$$

and

$$U = \{j \in T' \mid v_j < b_j \leq x_j\} \cup \{-j \in T' \mid x_j \leq a_j < v_j\}.$$

Clearly, $T \cup U \subset T'$, while $T \cap U = \emptyset$. Moreover, if $U = \emptyset$ we have that $A'(T, U) = A'(T)$, while $A'(\emptyset) = \{v\}$. The K'-triangulation subdivides the region $A'(T, U)$ into t-simplices $\sigma(w^1, \pi(T))$ with $\pi(T)$ a permutation $(\pi_1, \ldots, \pi_t)$ of the t elements of T and with vertices $w^1, \ldots, w^{t+1}$ where v^1 is a grid point in $A'(T, U)$ and $w^{i+1} = w^i + De(\pi_i)$, $i = 1, \ldots, t$.

The piecewise linear path from v satisfying condition P is traced by a sequence of so-called (T, U)-complete simplices in $A'(T, U)$ for various T and U such that $T \cap U = \emptyset$ and $(T \cup U) \cap J(v) = \emptyset$, where

$$J(v) = \{i \mid v_i = b_i\} \cup \{-i \mid v_i = a_i\}.$$

The definition of a (T, U)-complete simplex is derived from a T'-complete simplex τ in $A(T')$ in the same way that condition P is derived from condition N. So, for some T', let $\tau(y^1, \ldots, y^{t'+1})$ be a T'-complete simplex $\tau(y^1, \gamma(T'))$ in $A(T')$ with respect to the function $g(z)$ in (2.4). From Section 2 we know that the $n+1$ system of linear equations

$$\sum_{h+1}^{t'+1} \lambda_h g'(y^h) + \sum_{i \in \underline{T}'} \mu_i e'(i) + \beta s'(T') = e'(n+1) \tag{4.1}$$

where $g'(z) = (g(z)^{\mathrm{T}}, 1)^{\mathrm{T}}$ has a solution $\lambda_h^* \geq 0$, $h = 1, \ldots, t+1$, $\beta^* \geq 0$, and $-\beta^* \leq \mu_i^* \leq \beta^*$, $i \in \underline{T}'$, while $\tau \in A(T')$ implies that for all x in τ, $x_j = v_j$ when $j \in \underline{T}'$, $x_j \geq v_j$ if $j \in T'$ and $x_j \leq v_j$ if $-j \in T'$. Now, for some x in the relative interior of τ, let U and T be the disjoint subsets of T' as defined above. Then the projection $p(\tau)$ of $\tau(y^1, \gamma(T'))$ on C is the t-simplex $\sigma(w^1, \pi(T))$ in $A'(T, U)$ with $w^1 = p(y^1)$ and with $\pi(T)$ the permutation of the t elements of T such that the components of $\pi(T)$ are in the same order of succession as they appear in $\gamma(T')$.

We now derive from (4.1) the system of linear equations corresponding to the t-simplex σ in $A'(T, U)$. Since $g(x)=f(p(x))+x-p(x)$ the first term of (4.1) can be written as

$$\sum_{h=1}^{t'+1} \lambda_h f'(p(y^h))+\sum_{h=1}^{t'+1} \lambda_h((y^h-p(y^h))^{\mathrm{T}}, 0)^{\mathrm{T}}. \tag{4.2}$$

From the definition of $A'(T, U)$ it follows that for all h there are nonnegative numbers $\alpha(i, h)$, $i\in T'\setminus T$, such that

$$y^h-p(y^h)=\sum_{i\in T'\setminus T} \alpha(i, h)e(i).$$

Recall that $e(i)=-e(-i)$ if $i<0$. With $\vartheta_j=\sum_{h\in H(j)} \lambda_h$ where $H(j)=\{h\,|\,p(y^h)=w^j\}$ the first term in (4.2) becomes

$$\sum_{j=1}^{t+1} \vartheta_j f'(w^j).$$

Furthermore, for $k\in T'\setminus T$ we define

$$\mu_k=\sum_h \lambda_h\alpha(k, h)\,\mathrm{sign}(k)+\beta s_k(T'\setminus T).$$

Then we can combine the second term of (4.2) with the second and third term of (4.1) to

$$\sum_{i\in T} \mu_i e'(i)+\beta s'(T).$$

This gives us the next definition of a (T, U)-complete simplex.

Definition 4.1. For T and U in Z, such that $T\cap U=\emptyset$, $T\cup U\in Z$, $v_j=b_j$ implies $j\notin T\cup U$ and $v_j=a_j$ implies $-j\notin T\cup U$, a t-simplex $\sigma(w^1, \ldots, w^{t+1})$, $t=|T|$, is (T, U)-complete if the system of $n+1$ linear equations

$$\sum_{j=1}^{t+1} \vartheta_j f'(w^j)+\sum_{h\in T} \mu_h e'(h)+\beta s'(T)=e'(n+1) \tag{4.3}$$

has a solution $\vartheta_1^*, \ldots, \vartheta_{t+1}^*$, μ_h^*, $h\in T$, and β^* satisfying
T1. $\vartheta_1^*, \ldots, \vartheta_{t+1}^*\geqslant 0$
T2. $\beta^*\geqslant 0$.
T3. $\mu_h^*\geqslant -\beta^*$ if $h\in \underline{T\cup U}$ and $v_h>a_h$.
T4. $\mu_h^*\leqslant \beta^*$ if $h\in \underline{T\cup U}$ and $v_h<b_h$.
T5. $\mu_h^*\geqslant \beta^*$ if $h\in U$.
T6. $\mu_h^*\leqslant -\beta^*$ if $-h\in U$.

The next lemma shows the relation between a (T, U)-complete simplex in $A'(T, U)$ and condition P.

Lemma 4.2. *A t-simplex* $\sigma(w^1, \dots, w^{t+1})$ *in* $A'(T, U)$ *is* (T, U)*-complete if and only if the point* $x = \sum_j \vartheta_j w^j$ *in* σ *satisfies condition P.*

Proof. Let $(x, F(x), \alpha)$ be a triple satisfying condition P so that $x \in A'(T, U)$, $T = \{j \mid v_j < x_j < b_j\} \cup \{-j \mid a_j < x_j < v_j\}$ and $U = \{j \mid x_j = b_j > v_j\} \cup \{-j \mid x_j = a_j < v_j\}$. Then σ is a (T, U)-complete t-simplex in $A'(T, U)$ with solution $\vartheta_j^*, j = 1, \dots, t+1$, μ_h^*, $h \in \underline{T}$, and β^* given by

$$\beta^* = \alpha/(1+\alpha), \qquad \vartheta_j^* = \vartheta_j(1-\beta^*), \qquad j = 1, \dots, t+1,$$
$$\mu_h^* = -F_h(x)(1-\beta^*), \qquad h \in \underline{T}. \tag{4.4}$$

Conversely, if $\sigma(w^1, \dots, w^{t+1})$ is a (T, U)-complete simplex in $A'(T, U)$ with solution $(\vartheta^*, \mu^*, \beta^*)$, then the triple $(x, F(x), \alpha)$ given by

$$x = \sum_j \vartheta_j^* w^j/(1-\beta^*), \qquad F_j(x) = -\beta^* s_j(T)/(1-\beta^*), \quad j \notin \underline{T},$$
$$F_h(x) = -\mu_h^*/(1-\beta^*), \quad h \in \underline{T}, \qquad \alpha = \beta^*/(1-\beta^*), \tag{4.5}$$

satisfies condition P.

Without loss of generality we can make the next nondegeneracy assumption.

Assumption 4.3. Any (T, U)-complete simplex $\sigma(w^1, \dots, w^{t+1})$ in $A'(T, U)$ has exactly two solutions (ϑ, μ, β) with just one of the constraints T1–T6 binding.

A solution with one of the constraints binding is called a basic solution. Each point on the line segment between these two basic solutions is also a solution to (4.3), but with none of the inequalities in T1–T6 binding. In fact, a line segment of solutions to (4.3) in a (T, U)-complete simplex in $A'(T, U)$ corresponds to a linear piece of points x in σ for which the triple $(x, F(x), \alpha)$ as given in (4.5) satisfies condition P. Consequently, the piecewise linear path of points x for which $(x, F(x), \alpha)$ satisfies condition P can be followed by moving in the system (4.3) from basic solution to basic solution with respect to a sequence of adjacent (T, U)-complete simplices in $A'(T, U)$ for varying feasible sets T and U. A (T, U)-complete simplex having a basic solution in which $\beta^* = 0$ is called a complete simplex. From T1–T6 and (4.5) it follows that for such a basic solution the point $x^* = \sum_j \vartheta_j^* w^j$ is an approximate solution to problem (2.3).

We now describe the steps of the algorithm to follow the sequence of (T, U)-complete adjacent simplices in $A'(T, U)$ for varying T and U from the zero-dimensional simplex $\{v\}$ in $A'(\emptyset, \emptyset)$ to a complete simplex. In case v lies not in C, the algorithm starts at $p(v)$. For some feasible sets T and U, let $\sigma(w^1, \dots, w^{t+1})$ be a (T, U)-complete simplex $\sigma(w^1, \pi(T))$ in $A'(T, U)$ generated by the algorithm and let x^1 and x^2 be the two points in σ corresponding to the two basic solutions of system (4.3). So, for both points x^1 and x^2 exactly one of the inequalities in T1–T6 is binding. Moving from x^1 to x^2 is nothing else than making a linear programming pivot step in the system of $n+1$ linear equations (4.3). When by this pivot step ϑ_h becomes zero for some index h, x^2 lies in the interior of the facet τ of σ opposite the vertex w^h so that τ is also (T, U)-complete. Then the vertex w^h of σ is replaced by the vertex w not in τ of the unique simplex σ' in $A'(T, U)$ sharing the facet τ with σ, unless τ lies in the boundary of $A'(T, U)$. The algorithm is continued by pivoting $f'(w)$ into the system of linear equations with respect to the new σ', etc. In this way, by alternating replacement steps and linear programming pivot steps, a sequence of adjacent (T, U)-complete simplices in $A'(T, U)$ is followed until β becomes zero, one of the inequalities in T3–T6 becomes binding or until a facet opposite to the vertex to be replaced lies in the boundary of $A'(T, U)$. In the first case a solution to (2.3) is found with respect to F. In the second case we have that at a basic solution to (4.3) $|\mu_h^*| = \beta^*$ for some $h \in \underline{T}$. Then, for $k = h$ or $-h$ depending on whether $\mu_h^* = \beta^*$ or $\mu_h^* = -\beta^*$ respectively, the simplex is also (T', U')-complete with $T' = T \cup \{k\}$, $U' = U$ if $h \in \underline{T \cup U}$ and $U' = U \backslash \{k\}$ if $k \in U$. The algorithm continues by replacing the columns $e'(h)$ and $s'(T)$ by $s'(T \cup \{k\})$, while a pivot step is made with $f'(w)$, where w is the vertex of the unique $(t+1)$-simplex in $A'(T', U')$ having σ as facet opposite w. This unique simplex is given in the next lemma. The proof of the lemma follows directly from the structure of the K'-triangulation, see e.g. [19].

Lemma 4.4. *Let $\sigma(w^1, \pi(T))$ be a (T, U)-complete t-simplex in $A'(T, U)$ with at a basic solution $|\mu_h^*| = \beta^*$ for some $h \in \underline{T}$ and let $k = h$ if $\mu_h^* = \beta^*$ and $k = -h$ if $\mu_h^* = -\beta^*$. Then the unique $(t+1)$-simplex τ in $A'(T', U')$ having σ as a facet is given by*

$$\tau = \tau(w^1, (\pi(T), k)) \qquad \text{if } h \in \underline{T \cup U}$$

$$\tau = \tau(w^1 - De(k), (k, \pi(T))) \qquad \text{if } k \in U.$$

Finally, let us consider the case that a t-simplex $\sigma(w^1, \pi(T))$ in $A'(T, U)$ has a (T, U)-complete facet τ in the boundary of $A'(T, U)$. Then for some $h \in T$ and with $k = |h|$ holds that for all x in τ either $x_k = v_k$ or $x_k = b_k > v_k$ and $h > 0$ or $x_k = a_k < v_k$ and $h < 0$. In the first case τ is also a (T', U')-complete $(t-1)$-simplex in $A'(T', U')$ with $T' = T \backslash \{h\}$ and $U' = U$. In the other two cases τ is a (T', U')-complete $(t-1)$-simplex in $A'(T', U')$ with $T' = T \backslash \{h\}$ and $U' = U \cup \{h\}$. The algorithm continues by generating (T', U')-complete simplices in $A'(T', U')$. To do so the column $s'(T)$ in the system (4.3) is replaced by $s'(T \backslash \{h\})$, while $e'(k)$ is reintroduced into the new system. The next lemma shows how to recognize that a facet τ of σ lies in the boundary of $A'(T, U)$. We refer again to [19] for the proof of a similar result.

Lemma 4.5. *Let $\sigma(w^1, \pi(T))$ be a t-simplex in $A'(T, U)$ having a (T, U)-complete facet τ in the boundary. Then either τ is the $(t-1)$-simplex $\tau(w^1, (\pi_1, \ldots, \pi_{t-1}))$ in $A'(T \backslash \{h\}, U)$ if τ lies opposite w^{t+1}, $\pi_t = h$ and $w^1_{|h|} = v_{|h|}$ or τ is $\tau(w^1 + De(h), (\pi_2, \ldots, \pi_t))$ in $A'(T \backslash \{h\}, U \cup \{h\})$ if τ lies opposite w^1, $\pi_1 = h$ and $w^2_h = b_h$ when $h > 0$ and $w^2_{|h|} = a_{|h|}$ when $h < 0$.*

The results given above show how the path of points satisfying condition P can be followed by a sequence of (T, U)-complete adjacent t-simplices in $A'(T, U)$ with varying T and U and how a change in the sets T and U can be performed. The formal steps of the algorithm are given in the appendix. Under Assumption 4.3 all steps of the algorithm are unique, so that no simplex can be generated more than once. Assuming that all the a_i's and b_i's are finite and hence that C is compact, the algorithm must terminate within a finite number of steps with a complete simplex yielding a solution to (2.3) with respect to the piecewise linear approximation F. By restarting the algorithm in a grid point close to this approximate solution point for a triangulation with a smaller mesh, the accuracy of the approximation can be improved.

If at least one of the a_i's is minus infinity or one of the b_i's is plus infinity, the path of generated simplices could go to infinity. The following theorem gives conditions guaranteeing that the path of simplices will be finite. These conditions are closely related to those of Eaves [3], Kojima [7] and Moré [17] (see also [9]).

Theorem 4.6. *Let I_- be the set of indices i such that a_i is minus infinity and let I_+ be the set of indices i such that b_i is plus infinity. Suppose that for all $i \in I_-$ there exists a $l_i < b_i$ such that $f_i(x) < 0$ when $x_i < l_i$ and that for all $i \in I_+$ there exists a $u_i > a_i$ such that $f_i(x) > 0$ when $x_i > u_i$, where $l_i < u_i$ if $i \in I_- \cup I_+$. Then the modified $2n$-ray algorithm for solving (2.3) with respect to F converges for any starting point v in C and for any grid size vector d.*

Now taking a sequence of triangulations of C with mesh going to zero, we obtain the next corollary.

Corollary 4.7. *Under the conditions of Theorem 4.6, problem (2.3) has a solution and each solution lies in the set C^* defined by*

$$C^* = \{x \in R^n \mid x_i \geq a_i \text{ when } i \notin I_-,\ x_i > l_i \text{ when } i \in I_-,$$
$$x_i \leq b_i \text{ when } i \notin I_+,\ x_i < u_i \text{ when } i \in I_+\}.$$

5. Concluding remarks

The $2n$-ray algorithm applied to solve (2.4) needs at least $n+1$ linear programming steps to find a completely labelled simplex. This number is generally lower for the modified algorithm described in the previous section. In fact, let w be the starting point for the orignal $2n$-ray algorithm and let $v = p(w)$ be the starting point for the modified algorithm. At the first cycle w will be chosen in C. Hence $v = p(w) = w$ and the number of linear programming steps will be the same for both algorithms providing that the zero point algorithm utilizes the partial linear structure of g

outside C. However, unless the solution point is in the interior of C, the found approximate solution to (2.4) lies in general outside C. So, except for the first cycle the starting point w will typically lie outside C and the starting point $v = p(w)$ of the modified algorithm on the boundary of C. In this case, the number of linear programming steps may differ a lot. The zero point finding algorithm still needs at least $n+1$ l.p. steps, whereas the modified algorithm needs at least $k+1$ l.p. steps, where k is the dimension of the face of C on which v lies. This is caused by the fact that due to the projection on C of the path followed by the $2n$-ray algorithm a sequence of generated adjacent cells or simplices outside C can reduce to just one (lower-dimensional) simplex in the boundary of C. In this way each time a whole sequence of l.p. pivot steps in (2.1) is reduced to only one step in (4.1). It may even occur that the new algorithm needs exactly $k+1$ steps and the zero point algorithm more than $n+1$. An example is sketched in Fig. 1, where we assume that $g_1(w) < -|g_2(w)|$. The modified algorithm finds the approximate solution x^* after 2 l.p. steps, with $f'(v)$ and $f'(z)$ respectively. The zero point algorithm first traces in $A(\{1\})$ a path from w to $\bar{w}$ since $g_1(w) < -|g_2(w)|$ and then from $\bar{w}$ to the approximate zero w^*. However, going from $\bar{w}$ to w^* the ray $A(\{2\}) = \{x \in R^2 | x_1 = w_1, x_2 \geq w_2\}$ has to be passed, which takes two additional l.p. steps, so that the total number of l.p. steps is 5. Observe that $x^* = p(w^*)$.

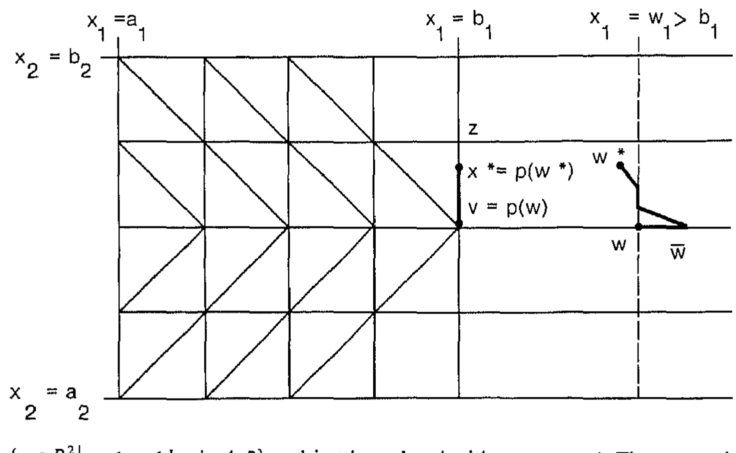

Fig. 1. $C = \{x \in R^2 | a_i \leq x_i \leq b_i,\ i = 1, 2\}$ and is triangulated with $m_1 = m_2 = 4$. The zero-point algorithm starts in w outside C and terminates after 5 iterations in w^*. The modified algorithm on C starts in $v = p(w)$ on the boundary of C and terminates after 2 iterations in $x^* = p(w^*)$.

Besides the $2n$-ray algorithm several other variable dimension algorithms have been developed to approximate a zero of a continuous function from R^n to R^n. Given some triangulation of R^n these algorithms generate from an arbitrarily chosen starting point v a path of adjacent simplices of varying dimensions. Each algorithm is characterized by the number of rays from which the starting point can be left. All these algorithms can be modified for solving (2.3) in the same way as described in this paper for the $2n$-ray algorithm so that also for these methods many l.p. steps can be saved. Especially the so-called 2^n-ray algorithm (see Wright [24]) is probably very attractive.

As done in [13] for the simplicial zero point algorithms on R^n also the modified algorithm can be interpreted as following a path of stationary points with respect to an expanding set being a convex polyhedron having as much as vertices as the number of rays of the algorithm. In addition we have to intersect the expanding set with the set C. Therefore the modified $2n$-ray algorithm follows the piecewise linear path of points $x(t)$ which starts at $x(0)=v$ satisfying, for $t \geq 0$,

$$x(t)^{\mathrm{T}}F(x(t)) \leq x^{\mathrm{T}}F(x(t))$$

for all x in $C \cap D(t)$ where

$$D(t)=\left\{x \in R^n \,\middle|\, \sum_i |x_i - v_i| \leq t\right\}.$$

As soon as for some t^* the point $x(t^*)$ lies in the interior of $D(t^*)$ we have $x(t)=x(t^*)$ for all $t>t^*$ and $x(t^*)$ solves problem (2.3) with respect to F. In case of the 2^n-ray algorithm the expanding set is equal to

$$D(t)=\{x \in R^n \mid \max_i |x_i - v_i| \leq t\}.$$

Appendix. The steps of the algorithm

Step 0. Set $T=\emptyset$, $U=\emptyset$, $\pi(T)=\emptyset$, $t=0$, $w^1=v$, $\sigma=\sigma(w^1, \pi(T))$, $p=1$.

Step 1. Calculate $f'(w^p)$ and perform a linear programming pivot step by bringing $f'(w^p)$ into the system of $n+1$ linear equations

$$\sum_{i \neq p} \lambda_i f'(w^i) + \sum_{h \in \underline{T}} \mu_h e'(h) + \beta' s(T) = e'(n+1).$$

Step 2. When β becomes zero, the algorithm terminates and $\sum_i \lambda_i w^i$ is an approximate solution to problem (2.3). When μ_h becomes $-\beta$ for some $h \in \underline{T}$, go to step 4. When μ_h becomes β for some $h \in \underline{T}$, go to step 5. Otherwise, λ_q becomes zero for some unique index q, $q \neq p$, and go to step 3.

Step 3. When $q=t+1$ and $w_h^1=v_h$ with $h=|\pi_t|$, go to step 6. When $q=1$ and, with $h=|\pi_1|$, $w_h^1=b_h-d_h$ if $\pi_1>0$, or $w_h^1=a_h+d_h$ if $\pi_1<0$, go to step 7. Otherwise, adapt w^1 and $\pi(T)$ according to Table 1 by replacing w^q, and return to step 1 with p the index of the new vertex of $\sigma(w^1, \pi(T))$.

Table 1
q is the index of the vertex of $\sigma(w^1, \pi(T))$ to be replaced

	w^1 becomes	$\pi(T)=(\pi_1, \ldots, \pi_t)$ becomes
$q=1$	$w^1+De(\pi_1)$	$(\pi_2, \ldots, \pi_t, \pi_1)$
$1<q<t+1$	w^1	$(\pi_1, \ldots, \pi_q, \pi_{q-1}, \ldots, \pi_t)$
$q=t+1$	$w^1-De(\pi_t)$	$(\pi_t, \pi_1, \ldots, \pi_{t-1})$

Step 4. Adapt the current system of linear equations by introducing $s'(T\cup\{-h\})$ and eliminating $e'(h)$ and $s'(T)$. When $h\in \underline{T\cup U}$, set $T=T\cup\{-h\}$ and $\pi(T)=(\pi_1,\ldots,\pi_t,-h)$. When $-h\in U$, set $w^1=w^1-De(-h)$, $T=T\cup\{-h\}$, $U=U\backslash\{-h\}$ and $\pi(T)=(-h,\pi_1,\ldots,\pi_t)$. Set $t=t+1$, $\sigma=\sigma(w^1,\pi(T))$, and return to step 1 with p the index of the new vertex of σ.

Step 5. Adapt the current system of linear equations by introducing $s'(T\cup\{h\})$ and eliminating $e'(h)$ and $s'(T)$. When $h\in \underline{T\cup U}$, set $T=T\cup\{h\}$ and $\pi(T)=(\pi_1,\ldots,\pi_t,h)$. When $h\in U$, set $w^1=w^1-De(h)$, $T=T\cup\{h\}$, $U=U\backslash\{h\}$ and $\pi(T)=(h,\pi_1,\ldots,\pi_t)$. Set $t=t+1$, $\sigma=\sigma(w^1,\pi(T))$, and return to step 1 with p the index of the new vertex of σ.

Step 6. Let $k=\pi_t$ and adapt the current system of linear equations by introducing $s'(T\backslash\{k\})$ and $e'(h)$ and eliminating $s'(T)$. Set $T=T\backslash\{k\}$, $\pi(T)=(\pi_1,\ldots,\pi_{t-1})$, $\sigma=\sigma(w^1,\pi(T))$, $t=t-1$ and perform a linear programming pivot step by decreasing μ_h from β when $k>0$ and increasing μ_h from $-\beta$ when $k<0$ in the system of linear equations

$$\sum_{i=1}^{t+1}\lambda_i f'(w^i)+\sum_{h\in T}\mu_h e'(h)+\beta s'(T)=e'(n+1).$$

Return to step 2.

Step 7. Let $k=\pi_1$ and adapt the current system of linear equations by introducing $s'(T\backslash\{k\})$ and $e'(h)$ and eliminating $s'(T)$. Set $T=T\backslash\{k\}$, $U=U\cup\{k\}$, $\pi(T)=(\pi_2,\ldots,\pi_t)$, $w^1=w^1+De(k)$, $\sigma=\sigma(w^1,\pi(T))$, $t=t-1$ and perform a linear programming pivot step by increasing μ_h from β when $k>0$ and decreasing μ_h from $-\beta$ when $k<0$ in the system of linear equations

$$\sum_{i=1}^{t+1}\lambda_i f'(w^i)+\sum_{h\in T}\mu_h e'(h)+\beta s'(T)=e'(n+1).$$

Return to step 2.

Acknowledgement

The authors express their thanks to the referees for their constructive remarks on an earlier version of this paper.

References

[1] E.L. Allgower and K. Georg, "Simplicial and continuation methods for approximating fixed points and solutions to systems of equations," *SIAM Review* 22 (1980) 28-85.

[2] S.N. Chow, J. Mallet-Paret and J.A. Yorke, "Finding zeroes of maps: homotopy methods that are constructive with probability one," *Mathematics of Computation* 32 (1978) 887-899.

[3] B.C. Eaves, "On the basic theory of complementarity," *Mathematical Programming* 1 (1971) 68-75.

[4] M.L. Fisher and F.J. Gould, "A simplicial algorithm for the nonlinear complementarity problem," *Mathematical Programming* 6 (1974) 281-300.

[5] C.B. Garcia, "The complementarity problem and its application," Ph.D. Thesis, Rensselaer Polytechnic Institute, Troy, NY (1973).
[6] G.J. Habetler and K.M. Kostreva, "On a direct algorithm for nonlinear complementarity problems," *SIAM Journal of Control and Optimization* 16 (1978) 504–511.
[7] M. Kojima, "Computational methods for solving the nonlinear complementarity problem," Keio Engineering Reports 27, Keio University, Yokohama, Japan (1974).
[8] M. Kojima and R. Saigal, "On the number of solutions to a class of complementarity problems," *Mathematical Programming* 21 (1981) 190–203.
[9] M. Kojima and Y. Yamamoto, "Variable dimension algorithms: Basic theory, interpretations and extensions of some existing methods," *Mathematical Programming* 24 (1982) 177–215.
[10] M. Kojima and Y. Yamamoto, "A unified approach to the implementation of several restart fixed point algorithms and a new variable dimension algorithm," *Mathematical Programming* 28 (1984) 288–328.
[11] G. van der Laan and A.J.J. Talman, "A restart algorithm for computing fixed points without an extra dimension," *Mathematical Programming* 17 (1979) 74–84.
[12] G. van der Laan and A.J.J. Talman, "A class of simplicial restart fixed point algorithms without an extra dimension," *Mathematical Programming* 20 (1981) 33–48.
[13] G. van der Laan and A.J.J. Talman, "Simplicial algorithms for finding stationary points, a unifying description," *Journal of Optimization Theory and Applications* 50 (1986) 165–182.
[14] H.J. Lüthi, "A simplicial approximation of a solution for the nonlinear complementarity problem," *Mathematical Programming* 9 (1975) 278–293.
[15] O.L. Mangasarian, "Equivalence of the complementarity problem to a system of nonlinear equations," *SIAM Journal of Applied Mathematics* 31 (1976) 89–92.
[16] O.H. Merrill, "Applications and extension of an algorithm that computes fixed points of certain upper semi-continuous point-to-set mappings," Ph.D. Thesis, University of Michigan, Ann Arbor, Mich. (1972).
[17] J.J. Moré, "Coercivity conditions in nonlinear complementarity algorithms," *SIAM Review* 16 (1974) 1–15.
[18] R. Saigal, "A homotopy for solving large, sparse and structured fixed point problems," *Mathematics of Operations Research* 8 (1983) 557–578.
[19] A.J.J. Talman, "Variable dimension fixed point algorithms and triangulations," Mathematical Centre Tracts 128 (Mathematisch Centrum, Amsterdam, 1980).
[20] 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 al., eds., *Homotopy Methods and Global Convergence* (Plenum Press, New York, 1983) pp. 267–286.
[21] M. J. Todd, "Improving the convergence of fixed point algorithms," *Mathematical Programming Study* 7 (1978) 151–169.
[22] M.J. Todd, "Global and local convergence and monotonicity results for a recent variable dimension simplicial algorithm," in: W. Forster, ed., *Numerical Solution of Highly Nonlinear Problems* (North-Holland, Amsterdam, 1980) pp. 43–69.
[23] L.T. Watson, "Solving the nonlinear complementarity problem by a homotopy method," *SIAM Journal of Control and Optimization* 17 (1979) 36–46.
[24] A.H. Wright, "The octahedral algorithm, a new simplicial fixed point algorithm," *Mathematical Programming* 21 (1981) 47–69.
[25] Y. Yamamoto, "A new variable dimension algorithm for the fixed point problem," *Mathematical Programming* 25 (1983) 329–342.