Tilburg University

A procedure for finding Nash equilibria in bi-matrix games

van den Elzen, A.H.; Talman, A.J.J.

Publication date:
1988

[Link to publication in Tilburg University Research Portal](#)

Citation for published version (APA):

van den Elzen, A. H., & Talman, A. J. J. (1988). *A procedure for finding Nash equilibria in bi-matrix games*. (Research memorandum / Tilburg University, Department of Economics; Vol. FEW 334). Unknown Publisher.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

2

CBM
R

7626
1988
334

UNIVERSITY

ALIEKE

UNIVERSITEIT
BRABANT

POSTBOX 90153
5000 LE TILBURG
THE NETHERLANDS

* C I N O 1 3 3 6 *

DEPARTMENT OF ECONOMICS
RESEARCH MEMORANDUM

1988

R7626

1988

n. 334

**A PROCEDURE FOR FINDING NASH EQUILIBRIA
IN BI-MATRIX GAMES**

A.H. van den Elzen
A.J.J. Talman

FEW 334

Department of Econometrics
Tilburg University
Tilburg, The Netherlands

R 57

618.25

A procedure for finding Nash equilibria in bi-matrix games

A.H. van den Elzen*, A.J.J. Talman

Abstract

In this paper we consider the computation of Nash equilibria for noncooperative bi-matrix games. The standard method for finding a Nash equilibrium in such a game is the Lemke-Howson method. That method operates by solving a related linear complementarity problem (LCP). However, the method may fail to reach certain equilibria because it can only start from a limited number of strategy vectors. The method we propose here finds an equilibrium by solving a related stationary point problem (SPP). Contrary to the Lemke-Howson method it can start from any strategy vector. Besides, the path of vectors along which the equilibrium is reached has an appealing game-theoretic interpretation. An important feature of the algorithm is that it finds a perfect equilibrium when at the start all actions are played with positive probability. Furthermore we can in principle find all Nash equilibria by repeated application of the algorithm starting from different strategy vectors.

Keywords: bi-matrix game, Nash-equilibrium, complementarity.

* This author is financially supported by the Co-operation Centre Tilburg and Eindhoven Universities, The Netherlands.

Department of Econometrics, Tilburg University, P.O. Box 90153, 5000 LE 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 and Sciences.

1. Introduction

In this paper we consider the problem of finding Nash equilibria in mixed strategies for a noncooperative bi-matrix game. A noncooperative bi-matrix game is a two-person game with a finite set of actions for each player. The payoffs of a player are listed in a matrix. The Nash equilibrium is the standard equilibrium concept for a noncooperative game. It states that a strategy is an equilibrium strategy when no player can improve upon his situation by deviating from his strategy while all other players keep on playing their strategies.

The standard method for solving a bi-matrix game is the Lemke-Howson method (see [3]). That method finds a Nash equilibrium by solving a related linear complementarity problem (LCP), which is not defined on the strategy space but on a nonnegative orthant. Each solution then uniquely determines an equilibrium on the strategy space.

The method proposed in this paper finds a Nash equilibrium by solving a stationary point problem (SPP) on the strategy space. Thus, contrary to the Lemke-Howson procedure it directly operates on the strategy space. Both methods are complementary pivoting algorithms and therefore find a positively indexed equilibrium (see [5]). However, because the choice of the starting vector for the Lemke-Howson procedure is restricted, that method may fail to reach certain positively indexed equilibria. In our method the choice of the starting vector is free and therefore we can find in principle all positively indexed equilibria by a repeated application of our algorithm from several starting vectors. However, the problem which remains is to find suitable starting vectors. In this paper we also show how our algorithm can find negatively indexed equilibria by restarting from the positively indexed equilibria already found. Our method can be seen as a strategy adjustment process having an appealing game-theoretic interpretation. In this respect it is also interesting that the algorithm finds a perfect equilibrium whenever the starting vector lies in the interior of the strategy space, i.e. when at the start all actions are played with a positive probability.

The organization of the paper is as follows. In Section 2 we show that the set of Nash equilibria in a bi-matrix game can be seen as the

solution set of a stationary point problem. Our solution procedure is presented in Section 3. In Section 4 we prove the perfectness of the equilibrium found by the procedure when starting from a strategy vector in the interior of the strategy space. In Section 5 we show how our algorithm can find negatively indexed equilibria. We also illustrate in an example that our method can find more equilibria than the Lemke-Howson method. Finally, some remarks are made concerning the computational speed of both algorithms.

2. Solving the bi-matrix game as a stationary point problem

A bi-matrix game is a tuple (n_1, n_2, A, B) in which $n_1(n_2)$ denotes the number of pure strategies of player 1(2) while $A(B)$ is the payoff matrix of player 1(2). The pure strategies of player 1(2) are indexed by $(1, k)$, $k=1, \dots, n_1$, $((2, k)$, $k=1, \dots, n_2$). The matrices A and B have dimension $n_1 \times n_2$. An element a_{jk} of A denotes the payoff to player 1 if he plays his j -th pure strategy while player 2 plays his k -th pure strategy. Similarly, b_{jk} denotes the payoff of player 2 in that situation.

The strategy space of player j , $j=1, 2$, is defined as the (n_j-1) -dimensional unit simplex $S_j^{n_j-1} := \{x_j \in \mathbb{R}_+^{n_j} \mid \sum_{k=1}^{n_j} x_{jk} = 1\}$. A vector x_j in $S_j^{n_j-1}$, $j=1, 2$, represents the mixed strategy of player j at which he plays his k -th pure strategy with probability x_{jk} . The n_j vertices of $S_j^{n_j-1}$ are the unit vectors in $\mathbb{R}^{n_j}$. They correspond to the pure strategies of player j , $j=1, 2$. The strategy space of the game, S , is equal to $S_1^{n_1-1} \times S_2^{n_2-1}$. A vector x in S is called a strategy vector and is equal to (x_1, x_2) with $x_1 \in S_1^{n_1-1}$ and $x_2 \in S_2^{n_2-1}$. Each vertex of S corresponds to a strategy vector at which both players play a pure strategy. The set S being the product of two unit simplices is called a simplotope.

On the set S we define the function $z: S \rightarrow \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ as follows. For each x in S the vector $z(x) = (z_1(x), z_2(x))$ with $z_1(x) \in \mathbb{R}^{n_1}$, $z_2(x) \in \mathbb{R}^{n_2}$, is defined by

$$z_1(x) = Ax_2 \text{ and } z_2(x) = B^T x_1. \quad (2.1)$$

The element $z_{1k}(x)$, $k=1, \dots, n_1$, is the payoff to player 1 when he plays his k -th pure strategy while player 2 plays x_2 . Similarly, $z_{2k}(x)$, $k=1, \dots, n_2$, is the payoff to player 2 when he plays his j -th pure strategy while player 1 plays x_1 . The number $x_j^T z_j(x)$ is the expected payoff to player j , $j=1,2$, when the strategy vector x is played.

A strategy vector x^* in S is called a Nash equilibrium if

$$x_j^T z_j(x^*) \leq x_j^{*T} z_j(x^*) \quad , \quad x_j \in S^{n_j-1}, j=1,2. \quad (2.2)$$

Thus, deviating from x_j^* will not increase the expected payoff to player j . The problem of finding an x^* in S such that x^* satisfies (2.2) is known as the stationary point problem (SPP) on S w.r.t. z . Since $x_j^T z_j(x^*)$ is linear in x_j we can restrict ourselves in (2.2) to the n_j vertices of S^{n_j-1} , $j = 1,2$, so that x^* in S is a Nash equilibrium if and only if $z_{jk}(x^*) \leq x_j^{*T} z_j(x^*)$, $k=1, \dots, n_j$, $j=1,2$. From this it is straightforward to derive that x^* is a Nash equilibrium if and only if for all k , $k=1, \dots, n_j$,

$$z_{jk}(x^*) \max_h z_{jh}(x^*) \text{ when } x_{jk}^* > 0, j=1,2. \quad (2.3)$$

Thus, at a Nash equilibrium only actions that are optimal for a player are played with positive probability.

Notice that at a Nash equilibrium x^* the expected payoff to player j is equal to $\max_h z_{jh}(x^*)$ since $z_{jk}(x^*) < \max_h z_{jh}(x^*)$ implies $x_{jk}^* = 0$.

3. The procedure

In this section we describe an algorithm for finding a solution to the SPP as given in (2.3). Starting from an arbitrarily chosen strategy vector $v = (v_1, v_2)$ in S , the algorithm generates a piecewise linear path of points in S leading from v to a Nash equilibrium. More precisely, points $x = (x_1, x_2) \in S$ on the path generated by the algorithm satisfy the following conditions. For $k=1, \dots, n_1$,

$$\begin{aligned} x_{1k} &= b(x, v) \cdot v_{1k} & \text{if } z_{1k}(x) < \max_h z_{1h}(x) \\ x_{1k} &\geq b(x, v) \cdot v_{1k} & \text{if } z_{1k}(x) = \max_h z_{1h}(x), \end{aligned}$$

while for $k=1, \dots, n_2$, (3.1)

$$\begin{aligned} x_{2k} &= b(x, v) \cdot v_{2k} & \text{if } z_{2k}(x) < \max_h z_{2h}(x) \\ x_{2k} &\geq b(x, v) \cdot v_{2k} & \text{if } z_{2k}(x) = \max_h z_{2h}(x), \end{aligned}$$

where $0 \leq b(x, v) := \min_{(j, h)} \left\{ \frac{x_{jh}}{v_{jh}} \mid v_{jh} > 0 \right\} \leq 1$.

Observe that $x = v$ satisfies (3.1) with $b(x, v) = 1$. Also each Nash-equilibrium x^* satisfies (3.1) with $b(x^*, v) = 0$ or with $v_{jk} = 0$ if $z_{jk}(x^*) < \max_h z_{jh}(x^*)$. In the latter case the non-optimal actions are already played with probability zero at the start. In both cases $x_{jk}^* = b(x^*, v) \cdot v_{jk}$ is equal to zero if (j, k) is a non-optimal action at x^* . Under some non-degeneracy condition the set of points satisfying (3.1) contains a piecewise linear path, P , from v to a Nash equilibrium. This path will be followed by the algorithm. The notion of non-degeneracy will be made precise further on, but it is for example required that at $x = v$ both $\max_k z_{1k}(x)$ and $\max_l z_{2l}(x)$ are attained for a unique index. Thus, at the starting strategy vector v each player has only one optimal action. Suppose these maxima are attained for the actions $(1, k_1)$ and $(2, k_2)$ respectively. Clearly, from v , along P , $b(x, v)$ must decrease from 1. Thus, according to (3.1), initially vectors x in S are generated such that all the x_{1k} , $k \neq k_1$, and x_{2k} , $k \neq k_2$, are relatively decreased ($x_{ik} = b(x, v)v_{ik}$, $(i, k) \neq (1, k_1), (2, k_2)$), while both x_{1k_1} and x_{2k_2} are increased in order to keep x_1 in $S_1^{n_1-1}$ and x_2 in $S_2^{n_2-1}$. This is done till $b(x, v)$ becomes 0 and a Nash equilibrium is reached, or till a point x is reached at which $z_{jk}(x) = z_{jk_j}(x)$ for some (j, k) , $k \neq k_j$. Then x_{jk} is also relatively increased. In general the algorithm generates strategy vectors x such that elements x_{jk}/v_{jk} , related to the indices (j, k) for which $z_{jk}(x) < \max_h z_{jh}(x)$, are minimal. As soon as one of these components of $z(x)$, say $z_{jk}(x)$, becomes equal to $\max_h z_{jh}(x)$, then x_{jk}/v_{jk} is increased. On the other hand, if a vector x is generated such that x_{jk}/v_{jk} with $z_{jk}(x) = \max_h z_{jh}(x)$ becomes minimal, i.e. equal to $b(x, v)$, then vectors y are generated with the (j, k) -th component equal to $b(y, v)v_{jk}$ and $z_{jk}(y) < \max_h z_{jh}(y)$. The latter guarantees convergence to an equilibrium.

To illustrate the foregoing let us provide a simple example of a game in which both players have two pure strategies and the matrices A and B are given by

$$A = \begin{bmatrix} 4 & 0 \\ -1 & 3 \end{bmatrix} \text{ and } B = \begin{bmatrix} 4 & -3 \\ 2 & 4 \end{bmatrix}.$$

The strategy space for this game is equal to $S = S^1 \times S^1 = \{x \in \mathbb{R}_+^4 \mid x_{11} + x_{12} = 1 \text{ and } x_{21} + x_{22} = 1\}$, and is displayed in Figure 3.1.

Figure 3.1. The path P of points satisfying (3.1) from v to a Nash equilibrium.

Consider the starting point $v = ((\frac{1}{8}, \frac{7}{8}), (\frac{1}{2}, \frac{1}{2}))$. It is straightforward to verify that $z(v) = ((2, 1), (\frac{9}{4}, \frac{25}{8}))$. Thus at v, $z_{11}(v)$ and $z_{22}(v)$ are maximal. So, the algorithm leaves v in the direction of $((1, 0), (0, 1))$, i.e. x_{11} and x_{22} are both increased. The algorithm continues in this way till the vector $a = ((\frac{2}{9}, \frac{7}{9}), (\frac{4}{9}, \frac{5}{9}))$ is reached at which $z(a) = ((\frac{16}{9}, \frac{11}{9}), (\frac{22}{9}, \frac{22}{9}))$, i.e. at $x = a$, $z_{21}(x)$ has become equal to $z_{22}(x)$. Observe that along the

segment (v,a) the number $b(x,v)$ decreases from 1 to $\frac{8}{9}$. Next the algorithm continues, according to (3.1), by increasing x_{21} relatively away from x_{12} while keeping $z_{21}(x)$ equal to $z_{22}(x)$. It is easy to verify that the latter holds along the line segment (a,c) . In fact, $z_{21}(x) = z_{22}(x) = \frac{22}{9}$ if $x_{11} = \frac{2}{9}$. At $c = ((\frac{2}{9}, \frac{7}{9}), (\frac{5}{9}, \frac{4}{9}))$ with $z(c) = ((\frac{20}{9}, \frac{7}{9}), (\frac{22}{9}, \frac{22}{9}))$ we still have $z_{21}(c) = z_{22}(c)$ and $z_{11}(c) > z_{12}(c)$. But observe that at $x = c$, $\frac{x_{22}}{v_{22}}$ has become equal to $b(x,v) = \frac{x_{12}}{v_{12}} = \frac{8}{9}$. When keeping $z_{22}(x)$ equal to $z_{21}(x)$ one would have to generate points x with $\frac{x_{22}}{v_{22}} < \frac{x_{12}}{v_{12}}$ which contradicts (3.1). In that case one would leave S at the vector $((\frac{2}{9}, \frac{7}{9}), (1,0))$. But instead the algorithm continues by keeping x_{22} relatively minimal while, according to (3.1), $z_{22}(x)$ is decreased from $\max_h z_{2h}(x) = z_{21}(x)$. In this way the algorithm reaches the vertex $((1,0), (1,0))$ which is a pure Nash equilibrium with $z((1,0), (1,0)) = ((4, -1), (4, -3))$.

The idea behind the procedure is to generate a sequence of vectors x along which the set T of actions (j,k) for which $z_{jk}(x) = \max_h z_{jh}(x)$ grows while the probabilities related to the other strategies are driven down to zero, since if they are zero, a Nash-equilibrium is found, as follows from (2.3). However, the set T does not need to grow monotonically (cf. point c in the foregoing example). This guarantees the convergence of the algorithm to a Nash equilibrium.

We now rewrite (3.1) into a form suited for implementing the algorithm. By substituting (2.1) we obtain that the process generates from v the piecewise linear path P of strategy vectors $x \in S$ satisfying for $k=1, \dots, n_1$,

$$\begin{aligned} x_{1k} &= bv_{1k} & \text{if } A^k x_2 < \beta_1 \\ x_{1k} &\geq bv_{1k} & \text{if } A^k x_2 = \beta_1, \end{aligned}$$

and for $k=1, \dots, n_2$,

$$\begin{aligned} x_{2k} &= bv_{2k} & \text{if } x_1^T B_k < \beta_2 \\ x_{2k} &\geq bv_{2k} & \text{if } x_1^T B_k = \beta_2, \end{aligned}$$

(3.2)

where $b := b(x, v)$, C^k and C_k respectively the k -th row and k -th column of a matrix C , $\beta_1 = \max_k A^k x_2$, and $\beta_2 = \max_k x_1^T B_k$.

Before going further we introduce some additional notation. The set of pure strategies of player j is denoted by $I(j)$, $j = 1, 2$. Thus, $I(j) = \{(j, 1), \dots, (j, n_j)\}$. The set $I = I(1) \cup I(2)$ denotes the set of all pure strategies in the game. For each x in (3.2) there is at least one set $T \subset I$ such that $x_{jk} \geq bv_{jk}$ and $z_{jk}(x) = \beta_j$ for all $(j, k) \in T$ while $z_{ih}(x) \leq \beta_i$ and $x_{ih} = bv_{ih}$ for all $(i, h) \notin T$. From this observation we obtain that the procedure generates for a sequence of subsets T of I with $T_j := T \cap I(j) \neq \emptyset$, starting at $x = v$, strategy-vectors x in S such that

$$\begin{aligned} x_{1k} &= bv_{1k} & \text{and } A^k x_2 &\leq \beta_1 & \text{if } (1, k) \notin T \\ x_{1k} &\geq bv_{1k} & \text{and } A^k x_2 &= \beta_1 & \text{if } (1, k) \in T, \end{aligned}$$

and

(3.3)

$$\begin{aligned} x_{2k} &= bv_{2k} & \text{and } x_1^T B_k &\leq \beta_2 & \text{if } (2, k) \notin T \\ x_{2k} &\geq bv_{2k} & \text{and } x_1^T B_k &= \beta_2 & \text{if } (2, k) \in T. \end{aligned}$$

We now show how the piecewise linear path P from v to a Nash equilibrium is generated by a sequence of linear programming ($\ell.p$) pivot steps in a system of linear equations obtained from (3.3). Denote the set of points x satisfying (3.3) for certain $T \subset I$ by $B(T)$. Thus, the algorithm reaches a Nash equilibrium via a (finite) sequence of sets $B(T)$, $T \subset I$. Each nonempty $B(T)$ is a line segment and we need one $\ell.p$ step to follow such a $B(T)$. The system of linear equations in which the $\ell.p$ pivot step is made is obtained from (3.3) by introducing slack variables for each inequality. The slack variables for the inequalities $A^h x_2 \leq \beta_1$, $(1, h) \notin T$, are denoted by μ_{1h} and those for $x_1^T B_h \leq \beta_2$, $(2, h) \notin T$, by μ_{2h} . The slack variables for the inequalities $x_{jk} \geq bv_{jk}$, $(j, k) \in T$, are denoted by λ_{jk} . Adding these slacks to (3.3) we obtain that x belongs to $B(T)$ iff there exist $\lambda_{jk} \geq 0$, $(j, k) \in T$, $\mu_{jk} \geq 0$, $(j, k) \notin T$, $0 \leq b \leq 1$, $\beta_j \in \mathbb{R}$, such that for $j=1, 2$,

$$x_j = bv_j + \sum_{(j,k) \in T_j} \lambda_{jk} e_j(k) \text{ with } \sum_{(j,k) \in T_j} \lambda_{jk} = 1-b, \quad (3.4)$$

while

$$Ax_2 + \sum_{(1,h) \in T} \mu_{1h} e_1(h) = e_1 \beta_1$$

and

$$B^T x_1 + \sum_{(2,h) \in T} \mu_{2h} e_2(h) = e_2 \beta_2.$$

(3.5)

Here $e_j(k)$ is the k -th unit vector in $\mathbb{R}^{n_j}$ while e_j is the vector of ones in $\mathbb{R}^{n_j}$, $j=1,2$.

Substituting (3.4) in (3.5) gives the system of linear equations

$$\begin{aligned} b \begin{bmatrix} Av_2 \\ B^T v_1 \\ 1 \\ 1 \end{bmatrix} + \sum_{(1,k) \in T} \lambda_{1k} \begin{bmatrix} \underline{0} \\ B_k^T \\ 1 \\ 0 \end{bmatrix} + \sum_{(2,k) \in T} \lambda_{2k} \begin{bmatrix} A_k \\ \underline{0} \\ 0 \\ 1 \end{bmatrix} + \sum_{(1,h) \in T} \mu_{1h} \begin{bmatrix} e_1(h) \\ \underline{0} \\ 0 \\ 0 \end{bmatrix} \\ + \sum_{(2,h) \in T} \mu_{2h} \begin{bmatrix} \underline{0} \\ e_2(h) \\ 0 \\ 0 \end{bmatrix} - \beta_1 \begin{bmatrix} \underline{0} \\ e_1 \\ 0 \\ 0 \end{bmatrix} - \beta_2 \begin{bmatrix} \underline{0} \\ e_2 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \underline{0} \\ \underline{0} \\ 1 \\ 1 \end{bmatrix}, \end{aligned} \quad (3.6)$$

where $\underline{0}$ denotes a vector of zeros of appropriate length. In the system (3.6) there are $n_1 + n_2 + 2$ equations and $n_1 + n_2 + 3$ variables. The last two equations of (3.6) reflect the property that $\sum_{(j,k) \in T_j} \lambda_{jk} = 1 - b$, $j=1,2$. These equations can be eliminated by substituting for $j=1,2$ one of the λ_{jk} 's, say λ_{jk_j} , by

$$\lambda_{jk_j} = 1 - b - \sum_{\substack{(j,k) \in T_j \\ k \neq k_j}} \lambda_{jk}. \quad (3.7)$$

Let T_j^1 be the set of indices defined by $T_j^1 = T_j \setminus \{(j, k_j)\}$ and let $T^1 = T_1^1 \cup T_2^1$. All this, together with substituting b by $1-b$, gives the following system of equations,

$$\begin{aligned}
& b' \begin{bmatrix} A_{k_2} & -Av_2 \\ B_{k_1}^T & -B^T v_1 \end{bmatrix} + \sum_{(1,k) \in T^1} \lambda_{1k} \begin{bmatrix} \underline{0} \\ B_k^T - B_{k_1}^T \end{bmatrix} + \sum_{(2,k) \in T^1} \lambda_{2k} \begin{bmatrix} A_k - A_{k_2} \\ \underline{0} \end{bmatrix} \\
& + \sum_{(1,h) \notin T} \mu_{1h} \begin{bmatrix} e_1(h) \\ \underline{0} \end{bmatrix} + \sum_{(2,h) \notin T} \mu_{2h} \begin{bmatrix} \underline{0} \\ e_2(h) \end{bmatrix} - \beta_1 \begin{bmatrix} e_1 \\ \underline{0} \end{bmatrix} \\
& - \beta_2 \begin{bmatrix} \underline{0} \\ e_2 \end{bmatrix} = \begin{bmatrix} -Av_2 \\ -B^T v_1 \end{bmatrix}. \tag{3.8}
\end{aligned}$$

The latter system has only $n_1 + n_2$ equations and $n_1 + n_2 + 1$ variables, and the variables must satisfy $\lambda_{jk} \geq 0$, $(j,k) \in T^1$, $\sum_{(j,k) \in T_j^1} \lambda_{jk} \leq b'$, $j=1,2$, $0 \leq b' \leq 1$, $\mu_{jh} \geq 0$, $(j,h) \notin T$. A solution to this system is denoted by $(b', \lambda_1, \lambda_2, \mu_1, \mu_2, \beta_1, \beta_2)$. The $\ell.p$ pivot steps of the algorithm are made in system (3.8).

We first consider the system (3.8) at the starting point $x = v$, assuming that the point v is not a Nash equilibrium vector. At v the index set T is equal to $T^0 = \{(1, k_1), (2, k_2)\}$ with k_1 and k_2 the (unique) indices for which

$$A^{k_1} v_2 = \max_k A^k v_2 \text{ and } v_1^T B_{k_2} = \max_k v_1^T B_k.$$

The corresponding solution of (3.8) is $b' = 0$, $\mu_{1h} = \beta_1 - A^h v_2$ for $h \neq k_1$, $\mu_{2h} = \beta_2 - v_1^T B_h$ for $h \neq k_2$, $\beta_1 = A^{k_1} v_2$, and $\beta_2 = v_1^T B_{k_2}$.

The algorithm leaves v by increasing b' from zero. According to (3.7) this means that λ_{1k_1} and λ_{2k_2} are increased from zero. Recall that this was just what happened in the example in the beginning of this section. In this way the algorithm follows $B(T^0)$ by making an $\ell.p$ pivot step in (3.8) with b' . As a result of this pivot step either b' becomes equal to one or μ_{ih} becomes zero for some (unique) index $(i,h) \notin T$. In the latter case z_{ih} becomes equal to z_{ik_i} and the algorithm continues by making a linear programming pivot step in (3.8) by increasing λ_{ih} from zero in order to follow the linear piece $B(T^0 \cup \{(i,h)\})$.

However, if $v_{jk} = 0$ for all $(j,k) \notin T^0 \cup \{(i,h)\}$ the algorithm terminates at the Nash equilibrium $\bar{x} = (\bar{x}_1, \bar{x}_2)$, where $\bar{x}_j = (1-b')v_j + b'e_j(k_j)$,

$j=1,2$. In the case that b' becomes one, i.e. $b = 0$, the algorithm stops at the pure Nash equilibrium $\bar{x} = (e_1(k_1), e_2(k_2))$. These observations follow from (2.3).

In general, the algorithm generates the linear piece $B(T)$ of the path P by making a linear programming pivot step in (3.8). We assume that for each such T at most one constraint on the variables is binding at a solution of (3.8).

Assumption 3.1. (nondegeneracy assumption). At each solution $(b', \lambda_1, \lambda_2, \mu_1, \mu_2, \beta_1, \beta_2)$ of (3.8) at most one of the constraints $0 \leq b' \leq 1$, $\lambda_{jk} \geq 0$ for $(j,k) \in T^1$, $b' \geq \sum_{k \neq k_j} \lambda_{jk}$, $\mu_{ih} \geq 0$ for $(i,h) \notin T$, is binding.

Assumption 3.1 is standard in linear programming. It assures that no more than one variable becomes zero at the same time. For example, it is not allowed that along the path several strategies become optimal simultaneously.

Furthermore, the solution set of (3.8) is bounded since T_j is nonempty for $j=1,2$. Indeed, since $\beta_j = \max_k z_{jk}(x)$ and z is a continuous function on the compact set S , β_j must be finite and therefore also the μ_{ih} with $(i,h) \notin T$. This together with Assumption 3.1 and the linearity of z , implies that the solution set of (3.8) for each T is either empty or a line segment with two endpoints. At each endpoint exactly one of the constraints is binding while no constraint is binding at interior solution points. The line segment of solutions to (3.8) corresponds according to (3.4) and (3.5) to $B(T)$. The algorithm moves from one endpoint to the other by relaxing the constraint that is binding at the first endpoint. The other endpoint is reached when another constraint becomes binding. More precisely, at that other endpoint, $(b', \lambda_1, \lambda_2, \mu_1, \mu_2, \beta_1, \beta_2)$, the following cases can happen

- i) $b' = 1$
- ii) $\lambda_{jk} = 0$ for some $(j,k) \in T^1$
- iii) $\sum_{(j,k) \in T_j^1} \lambda_{jk} = b'$ for some $j \in \{1,2\}$
- iv) $\mu_{ih} = 0$ for some $(i,h) \notin T$.

In case i) and in case iv) when additionally $v_{jk} = 0$ for all $(j,k) \notin T \cup \{(i,h)\}$, the point x as defined in (3.4) is a Nash solution. In case ii) the corresponding strategy vector $x = (x_1, x_2)$, being an endpoint of $B(T)$, is also an endpoint of $B(T \setminus \{(j,k)\})$. The algorithm then follows the line segment $B(T \setminus \{(j,k)\})$ by increasing the complementary variable μ_{jk} from zero in (3.8). Case iii) implies, according to (3.7), that $\lambda_{jk_j} = 0$ and can therefore be seen as a special case of ii). Thus the related vector x is also an endpoint of $B(T \setminus \{(j,k_j)\})$. In order to follow this line segment we have to adapt system (3.8) by eliminating another λ_{jk} . In principle any $\lambda_{jk'}$, $(j,k) \in T_j \setminus \{(j,k_j)\}$, can be taken. We suggest to take the largest $\lambda_{jk'}$, say $\lambda_{j\ell}$. Thus we then substitute $\lambda_{j\ell}$ by $b' - \sum_{\substack{(j,h) \in T_j \\ h \neq k_j, \ell}} \lambda_{jh}$.

This substitution can easily be performed in (3.8) by adding the column related to $\lambda_{j\ell}$ to the column related to b' and subtracting the same column from the columns related to λ_{jh} , $h \neq \ell$. The algorithm continues with this new system by increasing μ_{jk_j} from zero in order to follow $B(T \setminus \{(j,k_j)\})$.

Finally, if in case iv) the corresponding vector x is not a Nash equilibrium, then x is also an endpoint of $B(T \cup \{(i,h)\})$ and the algorithm follows this line segment by increasing the complementary variable λ_{ih} from zero.

In this way the algorithm generates by a sequence of complementary pivot steps the piecewise linear path P from v to a Nash-equilibrium. By a simple combinatorial argument it can be shown that the algorithm never cycles and always converges to a Nash equilibrium.

Let us apply the algorithm to the example presented earlier (see Figure 3.1.). At v for the solution $(b', \mu_1, \mu_2, \beta_1, \beta_2)$ of (3.8) holds that $\beta_1 = (Av_2)_1 = 2$, $\beta_2 = (B^T v_1)_2 = \frac{25}{8}$, $\mu_{12} = \beta_1 - (Av_2)_2 = 2 - 1 = 1$, $\mu_{21} = \beta_2 - (B^T v_1)_1 = \frac{25}{8} - \frac{9}{4} = \frac{7}{8}$, while $b' = 0$. The algorithm leaves v by increasing b' from zero. At a we have that $b' = \frac{1}{9}$, $\beta_1 = (Aa_2)_1 = \frac{16}{9}$, $\beta_2 = (B^T a_1)_2 = \frac{22}{9}$, $\mu_{12} = \frac{5}{9}$ while μ_{21} becomes zero. Thus at a the complementary variable of μ_{21} , i.e. λ_{21} , is increased from zero. The solution at c is $\beta_1 = (Ac_2)_1 = \frac{20}{9}$, $\beta_2 = (B^T c_1)_1 = (B^T c_1)_2 = \frac{22}{9}$, $\mu_{12} = \beta_1 - (Ac_2)_2 = \frac{13}{9}$, $\lambda_{21} = b'$ and $b' = \frac{1}{9}$. Then we are in case (iii), i.e., λ_{22} becomes zero. In

system (3.8), $(2, k_2)$ becomes $(2, 1)$ and the algorithm continues by increasing μ_{22} from zero. In the next step the algorithm reaches the Nash equilibrium $((1, 0), (1, 0))$ at which $b' = 1$ (case (i)).

We conclude this section by providing a game theoretic interpretation of the path generated by the procedure. The basic feature is that probabilities related to strategies not being optimal for a player are kept relatively (to v) minimal. At the starting vector v we have $z_{11}(v) > z_{12}(v)$ while $z_{21}(v) < z_{22}(v)$. Thus at v strategy $(1, 1)$ is the best reply for player 1 while strategy $(2, 2)$ is best for player 2. Now the algorithm increases from v the probabilities with which these strategies are played whereas the other probabilities are all decreased. The algorithm continues in this way till it generates the strategy vector a at which player 2 becomes in equilibrium ($z_{21}(a) = z_{22}(a)$). From a the algorithm generates strategy vectors x by relatively increasing probability x_{21} away from x_{12} while keeping player 2 in equilibrium. When strategy vector c is reached, the probability with which player 2 plays his second strategy has become relatively equal to c_{12} . Then the algorithm distorts the equilibrium situation of player 2. It continues by making for player 2 the strategy $(2, 2)$ less profitable than $(2, 1)$, i.e. $z_{22}(x)$ is made smaller than $z_{21}(x)$. Meanwhile, the probabilities x_{12} and x_{22} are kept relatively equal to each other and relatively smaller than x_{11} and x_{21} . In this way the Nash equilibrium $((1, 0), (1, 0))$ is reached.

4. Properties of the equilibrium found by the algorithm

In this section we discuss whether something can be said about game theoretic properties holding for the equilibrium found by our algorithm. More concrete, we would like to know whether the equilibrium found is isolated, quasi-strong, regular, essential, perfect, or proper. The precise definition of these concepts can be found for example in van Damme [2]. It turns out that we can only state that our algorithm finds a perfect equilibrium whenever it starts from an interior strategy vector. Other properties do not hold in general for the equilibrium reached by our procedure. The latter can easily be verified by applying our algorithm to simple examples.

Thus positive result is that our algorithm finds a perfect equilibrium whenever the starting point v is in the inner of the strategy space, i.e. $v_{jk} > 0$ for all $(j,k) \in I$. This is interesting the more because for bi-matrix games an equilibrium is perfect iff it is undominated (see Theorem 3.2.2 in [2]). To prove the statement it is most convenient to define a perfect equilibrium as a limit point of a sequence of ϵ -perfect equilibria. Thus first we have to define ϵ -perfectness.

Definition 4.1. The strategy vector $x(\epsilon) \in S$ is an ϵ -perfect equilibrium if it is completely mixed and satisfies:

$$\text{if } z_{jk}(x(\epsilon)) < z_{jl}(x(\epsilon)) \text{ then } x_{jk}(\epsilon) \leq \epsilon, \forall j,k,l.$$

Observe that an ϵ -perfect equilibrium need not to be a Nash-equilibrium. The concept only states that non-optimal pure reply strategies are played with a small probability, i.e. players make only small mistakes. Now x^* in S is a perfect equilibrium if x^* is a limit point of a sequence $\{x(\epsilon)\}_{\epsilon \downarrow 0}$, where for all ϵ , $x(\epsilon)$ is an ϵ -perfect equilibrium.

Theorem 4.2. If v lies in the interior of S then the algorithm finds a perfect Nash equilibrium.

Proof. From Section 3 we know that vectors x on the path generated by the algorithm satisfy

$$\begin{aligned} x_{jk} &= b(x,v) \cdot v_{jk} & \text{if } z_{jk}(x) < \max_l z_{jl}(x) \\ x_{jk} &\geq b(x,v) \cdot v_{jk} & \text{if } z_{jk}(x) = \max_l z_{jl}(x), \end{aligned}$$

$$\text{where } 0 \leq b(x,v) := \min_{(i,h)} \left\{ \frac{x_{ih}}{v_{ih}} \mid v_{ih} > 0 \right\} \leq 1.$$

Observe that all Nash equilibria x^* in the inner of S are perfect because then $z_{jk}(x^*) = \max_l z_{jl}(x^*)$ for all $(j,k) \in I$. Because $v_{jk} > 0$ for all $(j,k) \in I$, the algorithm can only reach an equilibrium x^* on the boundary of S if $b(x,v)$ decreases in the last iteration of the algorithm down to zero. Suppose $b(x,v)$ decreases along the line segment $[y, x^*]$ from $b(y,v)$ to zero. We define $I(x^*) := \{(j,k) \in I \mid x_{jk}^* = 0\}$ and (r,l) the index for

which $y_{rl} = \max_{(i,p) \in I(x^*)} y_{ip}$. Now each vector x on $[y, x^*)$ is an ϵ -perfect equilibrium $x(\epsilon)$ with $\epsilon = x_{rl}$. The limit point x^* of $\{x(\epsilon)\}_{\epsilon \downarrow 0}$ is perfect by definition. $\square$

Observe that we not only proved that our algorithm finds a perfect equilibrium but also that the last linear piece of the path consists of ϵ -perfect equilibria.

5. How to find more equilibria?

Both our algorithm and the Lemke-Howson procedure are complementary pivoting algorithms and therefore find positively indexed equilibria. The notion of the index of an equilibrium in a bi-matrix game is introduced by Shapley [5]. The index of an equilibrium is positive or negative depending on the sign of the determinant of a matrix related to the payoff structure at that equilibrium. For more details we refer the reader to [5]. In principle our algorithm can find all positively indexed equilibria. This because our procedure can start from almost all strategy vectors. Because the Lemke-Howson method can only start from a limited number of strategy vectors, it is possible that this procedure cannot detect some of the positively indexed equilibria. We illustrate this with an example given by Shapley in the article mentioned. Furthermore, we show that our algorithm can also find negatively indexed equilibria by restarting in a positively indexed equilibrium already found. This can only be done for games with an odd number of isolated equilibria, so that the number of negatively indexed equilibria is one less than the number of positively indexed equilibria (see [5]). This is not a great restriction because the set of bi-matrix games having these properties is dense in the set of all bi-matrix games (see [2]). We conclude this section with some remarks on the computational performance of the Lemke-Howson algorithm and our procedure.

Before giving the example of Shapley we first give a rough impression of the Lemke-Howson procedure. It solves a bi-matrix game (n_1, n_2, A, B) by solving a related linear complementarity problem on $\mathbb{R}^{n_1+n_2}$. Any vector $y = (y_1, y_2)$ in $\mathbb{R}_+^{n_1+n_2}$ corresponds to a strategy vector $x = (x_1, x_2)$ in S , where $x_j = y_j (\sum_{l=1}^{n_j} y_{jl})^{-1}$, $j = 1, 2$. By this transformation we can indicate

how the Lemke-Howson procedure operates on the strategy space S . It starts from a pure strategy vector at which player 1 plays a specific pure strategy while player 2 plays his best reply against that strategy. If this vector is not a Nash equilibrium, the algorithm starts with increasing the probability related to the best reply strategy of player 1 against the starting strategy of player 2. In this way, the Lemke-Howson method follows the best reply set of player 2 till it reaches a Nash equilibrium. Note that the starting vector is related to one of the pure strategies of player 1. By interchanging the players the maximal number of different starting vectors equals $n_1 + n_2$.

The example of Shapley concerns a 3×3 bi-matrix game with payoff matrices

$$A = \begin{bmatrix} 0 & 3 & 0 \\ 2 & 2 & 0 \\ 3 & 0 & 1 \end{bmatrix} \text{ and } B = \begin{bmatrix} 0 & 2 & 3 \\ 3 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Observe that the payoff structure for both players is identical. This game possesses two positively indexed Nash equilibria, i.e. $((0,0,1), (0,0,1))$ and $((\frac{1}{3}, \frac{2}{3}, 0), (\frac{1}{3}, \frac{2}{3}, 0))$. However, the Lemke-Howson algorithm can only find the first one. This because that method can only start in one of the three vectors $((1,0,0), (0,0,1))$, $((0,1,0), (1,0,0))$ and $((0,0,1), (0,0,1))$, each corresponding to a pure strategy of player 1 and the best reply of 2. Of course, interchanging the players gives no additional starting vectors because the payoffs are equal for both players.

With our algorithm we can also find the second positively indexed Nash equilibrium. For example, when it starts from $v = ((0,1,0), (0,1,0))$, it reaches $((\frac{1}{3}, \frac{2}{3}, 0), (\frac{1}{3}, \frac{2}{3}, 0))$ in one step. This because $z(v) = ((3,2,0), (3,2,0))$, making that the algorithm generates from v a path of vectors x with $x_{j1} \geq v_{j1}$, $x_{j2} \leq v_{j2}$ and $x_{j3} = 0$, $j = 1, 2$.

Next, we indicate how to find negatively indexed equilibria with our algorithm. We recall from Section 3 that our algorithm generates from a starting vector v a piecewise linear path of vectors x in S satisfying

$$\begin{aligned} x_{jk} &\geq b(x, v) \cdot v_{jk} & \text{if } z_{jk}(x) &= \max_h z_{jh}(x) \\ x_{jk} &= b(x, v) \cdot v_{jk} & \text{if } z_{jk}(x) &< \max_h z_{jh}(x), \end{aligned}$$

where $0 \leq b(x,v) = \min_{(i,h)} \left\{ \frac{x_{ih}}{v_{ih}} \mid v_{ih} > 0 \right\} \leq 1$.

Observe that these vectors all lie in the set $B_v := \cup_T B_v(T)$, $T \subsetneq I$, where

$$B_v(T) = \{x \in S \mid \begin{array}{l} x_{jk} \geq b(x,v) \cdot v_{jk} \text{ and } z_{jk}(x) = \max_h z_{jh}(x), (j,k) \in T \\ x_{jk} = b(x,v) \cdot v_{jk} \text{ and } z_{jk}(x) < \max_h z_{jh}(x), (j,k) \notin T \end{array}\}.$$

In general the set B_v is a 1-dimensional manifold. More precisely, it consists of disjoint piecewise linear paths. One such path connects v and a positively indexed Nash equilibrium while all other paths connect two Nash equilibria, one positively and one negatively indexed. In this way all other equilibria are connected.

How can we use the set B_v to find negatively indexed Nash equilibria? Suppose we have found k different positively indexed Nash equilibria by starting our algorithm from k different starting vectors. Next we consider the set B_v related to the starting vector v from which the first Nash equilibrium was found. In that set we successively restart our algorithm from the other $k-1$ positively indexed equilibria and find $k-1$ different negatively indexed equilibria by following the piecewise linear paths connecting the equilibria with positive and negative index. That this path is generated by our algorithm follows from the fact that vectors on the path satisfy the conditions (3.1). The only additional step is the initialization of the algorithm at a positively indexed equilibrium x^* . In practice this can be done by substituting v for the starting vector v^* related to x^* in the final system of the algorithm starting from v^* .

Again we illustrate this with the example of Section 3. Consider Figure 5.1. When we apply our algorithm starting from v^1 we find $((1,0), (1,0))$. Next we apply the algorithm from v^3 and find $((0,1), (0,1))$. The third (negatively indexed) equilibrium is found by restarting the algorithm in $((0,1), (0,1))$ while considering v^1 as starting vector in the system. Thus, in the final system related to $((0,1), (0,1))$ with $v = v^3$ we have to substitute v^1 for v^3 . Applying the algorithm in this manner means that we reach $((\frac{2}{9}, \frac{7}{9}), (\frac{3}{8}, \frac{5}{8}))$ by traversing the piecewise linear path in B_{v^1} connecting $((0,1), (0,1))$ and $((\frac{2}{9}, \frac{7}{9}), (\frac{3}{8}, \frac{5}{8}))$ in two steps.

Figure 5.1. The set B_{v^1} consists of a linear path connecting v^1 and $((1,0), (1,0))$ and a piecewise linear path connecting $((0,1), (0,1))$ and x^* . The line segment connecting $((0,1), (0,1))$ and a lies on the line through v^1 and $((0,1), (0,1))$ and equals $B_{v^1}(\{(1,2), (2,2)\})$. Furthermore, the line segment $[a, x^*]$ equals $B_{v^1}(\{(1,2), (2,1), (2,2)\})$.

To compare the computational strength of our algorithm and the Lemke-Howson procedure we applied both algorithms to several games of different dimension. More precisely, the number of actions of both players varied from 4 to 16. The computational speed was measured in the number of pivot steps needed to obtain an equilibrium. Overall seen, the Lemke-Howson method performs best. In particular that method is in favour for games of small dimension and games in which the number of actions of one player is small compared to the other. The Lemke-Howson method then operates on the best reply set of the player having the least number of actions. Our method is more suited for relatively large games with both players having a more or less equal number of actions. The reason is that our algorithm adapts all probabilities simultaneously. Of course, the

choice of the starting strategy vector is crucial for our algorithm. In general it seems impossible to select a 'best' starting vector from the data. We also did experiments with starting vectors related to rationalizable strategies (see [1] and [4]). At those starting vectors only actions are played that are a best reply against an action played by the other player. However, the results show no better results than those obtained for starting vectors at which all strategies are played with equal probability.

References

- [1] D. Bernheim, "Rationalizable strategic behaviour", *Econometrica* 52, 1984, 1007-1028.
- [2] E.E.C. van Damme, "Refinements of the Nash equilibrium concept", *Lecture Notes in Economics and Mathematical Systems* 219, Springer-Verlag, Berlin, 1983.
- [3] C.E. Lemke and J.T. Howson, "Equilibrium points of bimatrix games", *SIAM Journal of Applied Mathematics* 12, 1964, 413-423.
- [4] D. Pearce, "Rationalizable strategic behaviour and the problem of perfection", *Econometrica* 52, 1984, 1029-1040.
- [5] L.S. Shapley, "A note on the Lemke-Howson algorithm", *Mathematical Programming Studies* 1, 1974, 175-189.

IN 1987 REEDS VERSCHENEN

- 242 Gerard van den Berg
Nonstationarity in job search theory
- 243 Annie Cuyt, Brigitte Verdonk
Block-tridiagonal linear systems and branched continued fractions
- 244 J.C. de Vos, W. Vervaat
Local Times of Bernoulli Walk
- 245 Arie Kapteyn, Peter Kooreman, Rob Willemse
Some methodological issues in the implementation
of subjective poverty definitions
- 246 J.P.C. Kleijnen, J. Kriens, M.C.H.M. Lafleur, J.H.F. Pardoel
Sampling for Quality Inspection and Correction: AOQL Performance
Criteria
- 247 D.B.J. Schouten
Algemene theorie van de internationale conjuncturele en structurele
afhankelijkheden
- 248 F.C. Bussemaker, W.H. Haemers, J.J. Seidel, E. Spence
On (v,k,λ) graphs and designs with trivial automorphism group
- 249 Peter M. Kort
The Influence of a Stochastic Environment on the Firm's Optimal Dyna-
mic Investment Policy
- 250 R.H.J.M. Gradus
Preliminary version
The reaction of the firm on governmental policy: a game-theoretical
approach
- 251 J.G. de Gooijer, R.M.J. Heuts
Higher order moments of bilinear time series processes with symmetri-
cally distributed errors
- 252 P.H. Stevers, P.A.M. Versteijne
Evaluatie van marketing-activiteiten
- 253 H.P.A. Mulders, A.J. van Reeken
DATAAL - een hulpmiddel voor onderhoud van gegevensverzamelingen
- 254 P. Kooreman, A. Kapteyn
On the identifiability of household production functions with joint
products: A comment
- 255 B. van Riel
Was er een profit-squeeze in de Nederlandse industrie?
- 256 R.P. Gilles
Economies with coalitional structures and core-like equilibrium con-
cepts

- 257 P.H.M. Ruys, G. van der Laan
Computation of an industrial equilibrium
- 258 W.H. Haemers, A.E. Brouwer
Association schemes
- 259 G.J.M. van den Boom
Some modifications and applications of Rubinstein's perfect equilibrium model of bargaining
- 260 A.W.A. Boot, A.V. Thakor, G.F. Udell
Competition, Risk Neutrality and Loan Commitments
- 261 A.W.A. Boot, A.V. Thakor, G.F. Udell
Collateral and Borrower Risk
- 262 A. Kapteyn, I. Woittiez
Preference Interdependence and Habit Formation in Family Labor Supply
- 263 B. Bettonvil
A formal description of discrete event dynamic systems including perturbation analysis
- 264 Sylvester C.W. Eijffinger
A monthly model for the monetary policy in the Netherlands
- 265 F. van der Ploeg, A.J. de Zeeuw
Conflict over arms accumulation in market and command economies
- 266 F. van der Ploeg, A.J. de Zeeuw
Perfect equilibrium in a model of competitive arms accumulation
- 267 Aart de Zeeuw
Inflation and reputation: comment
- 268 A.J. de Zeeuw, F. van der Ploeg
Difference games and policy evaluation: a conceptual framework
- 269 Frederick van der Ploeg
Rationing in open economy and dynamic macroeconomics: a survey
- 270 G. van der Laan and A.J.J. Talman
Computing economic equilibria by variable dimension algorithms: state of the art
- 271 C.A.J.M. Dirven and A.J.J. Talman
A simplicial algorithm for finding equilibria in economies with linear production technologies
- 272 Th.E. Nijman and F.C. Palm
Consistent estimation of regression models with incompletely observed exogenous variables
- 273 Th.E. Nijman and F.C. Palm
Predictive accuracy gain from disaggregate sampling in arima - models

- 274 Raymond H.J.M. Gradus
The net present value of governmental policy: a possible way to find the Stackelberg solutions
- 275 Jack P.C. Kleijnen
A DSS for production planning: a case study including simulation and optimization
- 276 A.M.H. Gerards
A short proof of Tutte's characterization of totally unimodular matrices
- 277 Th. van de Klundert and F. van der Ploeg
Wage rigidity and capital mobility in an optimizing model of a small open economy
- 278 Peter M. Kort
The net present value in dynamic models of the firm
- 279 Th. van de Klundert
A Macroeconomic Two-Country Model with Price-Discriminating Monopolists
- 280 Arnoud Boot and Anjan V. Thakor
Dynamic equilibrium in a competitive credit market: intertemporal contracting as insurance against rationing
- 281 Arnoud Boot and Anjan V. Thakor
Appendix: "Dynamic equilibrium in a competitive credit market: intertemporal contracting as insurance against rationing"
- 282 Arnoud Boot, Anjan V. Thakor and Gregory F. Udell
Credible commitments, contract enforcement problems and banks: intermediation as credibility assurance
- 283 Eduard Ponds
Wage bargaining and business cycles a Goodwin-Nash model
- 284 Prof.Dr. hab. Stefan Mynarski
The mechanism of restoring equilibrium and stability in polish market
- 285 P. Meulendijks
An exercise in welfare economics (II)
- 286 S. Jørgensen, P.M. Kort, G.J.C.Th. van Schijndel
Optimal investment, financing and dividends: a Stackelberg differential game
- 287 E. Nijssen, W. Reijnders
Privatisering en commercialisering; een oriëntatie ten aanzien van verzelfstandiging
- 288 C.B. Mulder
Inefficiency of automatically linking unemployment benefits to private sector wage rates

- 289 M.H.C. Paardekooper
A Quadratically convergent parallel Jacobi process for almost diagonal matrices with distinct eigenvalues
- 290 Pieter H.M. Ruys
Industries with private and public enterprises
- 291 J.J.A. Moors & J.C. van Houwelingen
Estimation of linear models with inequality restrictions
- 292 Arthur van Soest, Peter Kooreman
Vakantiebestemming en -bestedingen
- 293 Rob Alessie, Raymond Gradus, Bertrand Melenberg
The problem of not observing small expenditures in a consumer expenditure survey
- 294 F. Boekema, L. Oerlemans, A.J. Hendriks
Kansrijkheid en economische potentie: Top-down en bottom-up analyses
- 295 Rob Alessie, Bertrand Melenberg, Guglielmo Weber
Consumption, Leisure and Earnings-Related Liquidity Constraints: A Note
- 296 Arthur van Soest, Peter Kooreman
Estimation of the indirect translog demand system with binding non-negativity constraints

IN 1988 REEDS VERSCHENEN

- 297 Bert Bettonvil
Factor screening by sequential bifurcation
- 298 Robert P. Gilles
On perfect competition in an economy with a coalitional structure
- 299 Willem Selen, Ruud M. Heuts
Capacitated Lot-Size Production Planning in Process Industry
- 300 J. Kriens, J.Th. van Lieshout
Notes on the Markowitz portfolio selection method
- 301 Bert Bettonvil, Jack P.C. Kleijnen
Measurement scales and resolution IV designs: a note
- 302 Theo Nijman, Marno Verbeek
Estimation of time dependent parameters in linear models
using cross sections, panels or both
- 303 Raymond H.J.M. Gradus
A differential game between government and firms: a non-cooperative
approach
- 304 Leo W.G. Strijbosch, Ronald J.M.M. Does
Comparison of bias-reducing methods for estimating the parameter in
dilution series
- 305 Drs. W.J. Reijnders, Drs. W.F. Verstappen
Strategische bespiegelingen betreffende het Nederlandse kwaliteits-
concept
- 306 J.P.C. Kleijnen, J. Kriens, H. Timmermans and H. Van den Wildenberg
Regression sampling in statistical auditing
- 307 Isolde Woittiez, Arie Kapteyn
A Model of Job Choice, Labour Supply and Wages
- 308 Jack P.C. Kleijnen
Simulation and optimization in production planning: A case study
- 309 Robert P. Gilles and Pieter H.M. Ruys
Relational constraints in coalition formation
- 310 Drs. H. Leo Theuns
Determinanten van de vraag naar vakantiereizen: een verkenning van
materiële en immateriële factoren
- 311 Peter M. Kort
Dynamic Firm Behaviour within an Uncertain Environment
- 312 J.P.C. Blanc
A numerical approach to cyclic-service queueing models

- 313 Drs. N.J. de Beer, Drs. A.M. van Nunen, Drs. M.O. Nijkamp
Does Morkmon Matter?
- 314 Th. van de Klundert
Wage differentials and employment in a two-sector model with a dual
labour market
- 315 Aart de Zeeuw, Fons Groot, Cees Withagen
On Credible Optimal Tax Rate Policies
- 316 Christian B. Mulder
Wage moderating effects of corporatism
Decentralized versus centralized wage setting in a union, firm,
government context
- 317 Jörg Glombowski, Michael Krüger
A short-period Goodwin growth cycle
- 318 Theo Nijman, Marno Verbeek, Arthur van Soest
The optimal design of rotating panels in a simple analysis of
variance model
- 319 Drs. S.V. Hannema, Drs. P.A.M. Versteijne
De toepassing en toekomst van public private partnership's bij de
grote en middelgrote Nederlandse gemeenten
- 320 Th. van de Klundert
Wage Rigidity, Capital Accumulation and Unemployment in a Small Open
Economy
- 321 M.H.C. Paardekooper
An upper and a lower bound for the distance of a manifold to a nearby
point
- 322 Th. ten Raa, F. van der Ploeg
A statistical approach to the problem of negatives in input-output
analysis
- 323 P. Kooreman
Household Labor Force Participation as a Cooperative Game; an Empiri-
cal Model
- 324 A.B.T.M. van Schaik
Persistent Unemployment and Long Run Growth
- 325 Dr. F.W.M. Boekema, Drs. L.A.G. Oerlemans
De lokale produktiestructuur doorgelicht.
Bedrijfstakingverkenningen ten behoeve van regionaal-economisch onder-
zoek
- 326 J.P.C. Kleijnen, J. Kriens, M.C.H.M. Lafleur, J.H.F. Pardoel
Sampling for quality inspection and correction: AOQL performance
criteria

- 327 Theo E. Nijman, Mark F.J. Steel
Exclusion restrictions in instrumental variables equations
- 328 B.B. van der Genugten
Estimation in linear regression under the presence of heteroskedasticity of a completely unknown form
- 329 Raymond H.J.M. Gradus
The employment policy of government: to create jobs or to let them create?
- 330 Hans Kremers, Dolf Talman
Solving the nonlinear complementarity problem with lower and upper bounds
- 331 Antoon van den Elzen
Interpretation and generalization of the Lemke-Howson algorithm
- 332 Jack P.C. Kleijnen
Analyzing simulation experiments with common random numbers, part II: Rao's approach
- 333 Jacek Osiewalski
Posterior and Predictive Densities for Nonlinear Regression.
A Partly Linear Model Case

Bibliotheek K. U. Brabant

17 000 01065983 8