Journal of the Operations Research
Society of Japan
Vol. 34, No. 2, June 1991

SIMPLICIAL ALGORITHM FOR COMPUTING A CORE ELEMENT IN A BALANCED GAME

Kazuya Kamiya
Osaka University

Dolf Talman
Tilburg University

(Received March 26, 1990)

Abstract In this paper we propose a simplicial algorithm to find a core element for balanced games without side payments. The algorithm subdivides an appropriate simplex into smaller simplices and generates from an arbitrarily chosen point a sequence of adjacent simplices of variable dimension. Within a finite number of iterations the algorithm finds a simplex yielding an approximating core element. If the accuracy of approximation is not satisfactory, the algorithm can be restarted with a smaller mesh size in order to improve the accuracy.

1. Introduction

It is well-known that a cooperative game need not to have an outcome which cannot be improved upon by any subset of players. In case of a game with side payments the core, consisting of all outcomes which cannot be improved, is nonempty if and only if the game is balanced, see Bondareva [4] and Shapley [8]. A core element in a balanced game with side payments can be easily calculated by solving a sequence of linear programming problems.

Games without side payments were introduced by Aumann and Peleg [2], and Aumann [1] developed the core concept for such games. Scarf [7] proved the nonemptiness of the core for such a game if it is balanced. Scarf gave a constructive proof based on the complementary pivoting technique introduced by Lemke and Howson [6]. Shapley [9] generalized the well-known Knaster-Kuratowski-Mazurkiewicz Theorem on the unit simplex in order to give a constructive proof of the nonemptiness of the core. In an arbitrary subdivision of the (unit) simplex into simplices, a sequence of adjacent simplices is generated, which is initiated at one of the corners of the big simplex. The terminal simplex yields an approximating core element.

In this paper we propose a simplicial algorithm which can be initiated at any point of the unit simplex. From that point the algorithm generates a sequence of adjacent simplices of varying dimension. The algorithm leaves the starting point along one out of $2^n - 2$ directions in case there are n players. This number corresponds to the number of proper coalitions in the game. The algorithm is based on the simplicial algorithm developed by Doup, van der Laan and Talman [5] for computing economic equilibria. Along the path of simplices generated by the algorithm, coalitions are added and sometimes deleted until a balanced set of coalitions has been found. Once such a set is obtained, an approximating core element has been found. If the accuracy of approximation at that point is not satisfactory, the algorithm can be restarted at that point with a smaller mesh size of the triangulation in order to improve the accuracy. Within a finite number of restarts any accuracy of approximation can be reached.

In Section 2 we describe a balanced game and define the core. Section 3 gives the steps of the algorithm for finding a core element. Concluding remarks are made in Section 4.

2. Balanced game and core

Let N denote the set $\{1, \cdots, n\}$ and 2^N the set of all nonempty subsets of N. We call the elements of N and the elements of 2^N, players and coalitions, respectively. A game is a pair (N, v) where v is a mapping from 2^N to the set of subsets of the n-dimensional euclidean space, R^N. The set $v(S)$ represents the set of payoff or utility vectors that the players of coalition S can ensure by themselves, regardless of the actions of players outside the coalition. For S in 2^N, let R^S denote the $|S|$-dimensional subspace of R^N with coordinates indexed by the elements of S. If $x \in R^N$ and $S \in 2^N$, then $x^S \in R^S$ will denote the projection of x on R^S.

Assumption 2.1: For each $S \in 2^N$, the set $v(S)$ satisfies
- i) if $x \in v(S)$ and $x_i = y_i$ for all $i \in S$, then $y \in v(S)$,
- ii) if $x \in v(S)$ and $y \leq x$, then $y \in v(S)$,
- iii) $v(S)$ is closed,
- iv) $\{x^S \mid x \in V(S)\}$ is nonempty and bounded from above.

Without loss of generality we assume that each set $v(\{i\}), i \in N$, has been normalized to the half space $\{x \mid x_i \leq 0\}$ and that the other $v(S)$'s have been shifted accordingly. The core of a game represents the set of feasible utility vectors that cannot be improved upon by any coalition.

Definition 2.1: The core of the game (N, v) is the set $C(N, v) = \{x \in v(N) \mid \nexists S \in 2^N$ and $y \in v(S)$ such that $y_i > x_i$ for all $i \in S\}$.

Under Assumption 2.1, the core is a closed and bounded set but may, however, be empty. It is a well-known fact that every balanced game has a nonempty core.

Let B be a collection of nonempty subsets of 2^N, and let $B_i = \{S \in B \mid i \in S\}$. The set B is said to be balanced if there exist nonnegative numbers δ_S, $S \in B$, such that

$$\sum_{S \in B_i} \delta_S = 1 \quad \text{for all } i \in N.$$

A game (N, v) is said to be balanced if for every balanced set B

$$\bigcap_{S \in B} v(S) \subset v(N).$$

Theorem 2.1 (Scarf [7]): Every balanced game has a nonempty core.

Let U be the $(n-1)$-dimensional subset of R^N defined by $U = \text{conv}\{-Mne(j) \mid j = 1, \cdots, n\}$ where $e(j)$ is the j-th unit vector in R^N and the number $M > 0$ is such that $x \in v(S)$ implies $x_i < M$ for every $i \in S$. Let e be the n-vector of ones. The function $\tau : U \to R_+$ is defined by

$$\tau(u) = \max\{r \in R \mid u + re \in \bigcup_{S \subset N} v(S)\}.$$

Clearly, τ is a continuous function on U, for example see Berge [3]. For $S \in 2^N$ we now define the set C_S by

$$C_S = \{u \in U \mid u + \tau(u)e \in v(S)\}.$$

Since $v(S)$ is closed, the set C_S is also closed. The algorithm will compute a point u^* in U such that for some balanced collection B^*

$$u^* \in \bigcap_{S \in B^*} C_S.$$

Then $x^* = u^* + \tau(u^*)e \in \bigcap_{S \in B^*} v(S) \subset v(N)$ and x^* lies in the core since x^* lies on the (upper) boundary of $\bigcup_{S \subset N} v(S)$.

Lemma 2.2 (Shapley [9]): For all $u \in U$, if $u \in C_S$ then $S \subset \{i \in N \mid u_i \neq 0\}$.

Proof: Let $u \in C_S$ and $T = \{i \in N \mid u_i \neq 0\}$. The lemma is trivial if $T = N$. So assume that $|T| < n$. Because $u_i = 0$ for all $i \notin T$, we have $\sum_{i \in T} u_i = -Mn$, so there exists a $k \in T$ for which $u_k < -M$. Since $u + \tau(u)e \in R^N_+$, we have $u_k + \tau(u) \geq 0$, and hence $\tau(u) > M$. On the other hand, $u + \tau(u)e \in v(S)$, so for every $j \in S, u_j + \tau(u) < M$. Therefore $u_j < 0$ for every $j \in S$, from which it follows that $S \subset T$.

Q.E.D.

The lemma will guarantee the algorithm never hits the boundary of the set U.

3. The algorithm

To describe the algorithm, let p be an arbitrarily chosen starting point in the relative interior of U. Next, let s be a sign vector in R^N, i.e., $s_j \in \{0, -1, +1\}$ for all $j \in N$. We call a sign vector s feasible if s contains at least one -1 and one $+1$. For a feasible sign vector s let the subset $A(s)$ of U be defined by

$$A(s) = \{u \in U \mid u_j/p_j = \max_h u_h/p_h \quad \text{if } s_j = -1$$

$$u_j/p_j = \min_h u_h/p_h \quad \text{if } s_j = +1\}.$$

Clearly, the dimension of $A(s)$ is equal to $t = |I^0(s)| + 1$ where

$$I^0(s) = \{i \in N \mid s_i = 0\}.$$

In particular, if the sign vector s does not contain zeros then $A(s)$ is a 1-dimensional set, being the line segment connecting p and the point $p(s)$ in the boundary of U given by $p_j(s) = 0$ for all j with $s_j = +1$ and $p_j(s) = -Mnp_j/\sum_{s_h=-1} p_h$ for all j with $s_j = -1$. For $n = 3$ the subdivision of U into sets $A(s)$ for an arbitrary p is illustrated in Figure 3.1. Next U is subdivided into $(n-1)$-dimensional simplices such that each $A(s)$ is triangulated into t-dimensional simplices, for example see Doup, van der Laan and Talman [5]. A t-dimensional simplex or t-simplex σ can be represented by its $t+1$ vertices $w^1, \cdots, w^{t+1}$. To each vertex w of the simplicial subdivision we assign a vector label $a(S)$ corresponding to some fixed coalition S for which w lies in C_S, where $a_j(S) = 1 - |S|/n$ for $j \in S$ and $a_j(S) = -|S|/n$ for $j \notin S$. Notice that $\sum_{j=1}^{n} a_j(S) = 0$. For $g = t$ or $t-1$, let $\sigma(w^1, \cdots, w^{g+1})$ be a g-simplex with vertices $w^1, \cdots, w^{g+1}$ in $A(s)$ for some feasible sign vector s. Let $a(S^j)$ be the vector label of vertex w^j, then we call σ s-complete if the system of linear equations

$$\sum_{j=1}^{g+1} \lambda_j \begin{bmatrix} a(S^j) \\ 1 \end{bmatrix} - \sum_{s_h \neq 0} \mu_h s_h \begin{bmatrix} e(h) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{3.1}$$

has a nonnegative solution $\lambda_j^*, j = 1, \cdots, g+1, \mu_h^*$ for $h \notin I^0(s)$. In particular, for $t = 1$ and $g = 0$, the zero-dimensional simplex consisting of the point p is s^0-complete with $s_i^0 = +1$ if $i \in S^0$ and $s_i^0 = -1$ if $i \notin S^0$, where S^0 is such that $a(S^0)$ is the vector label of p. If S^0 equals N, the point $p + \tau(p)e$ lies in the core. Suppose now that S^0 unequals N. Clearly, s^0 is feasible and does not contain zeros. Notice that there are $2^n - 2$ feasible sign vectors not containing zeros and that each such sign vector corresponds in this way to one of the $2^n - 2$ proper coalitions.

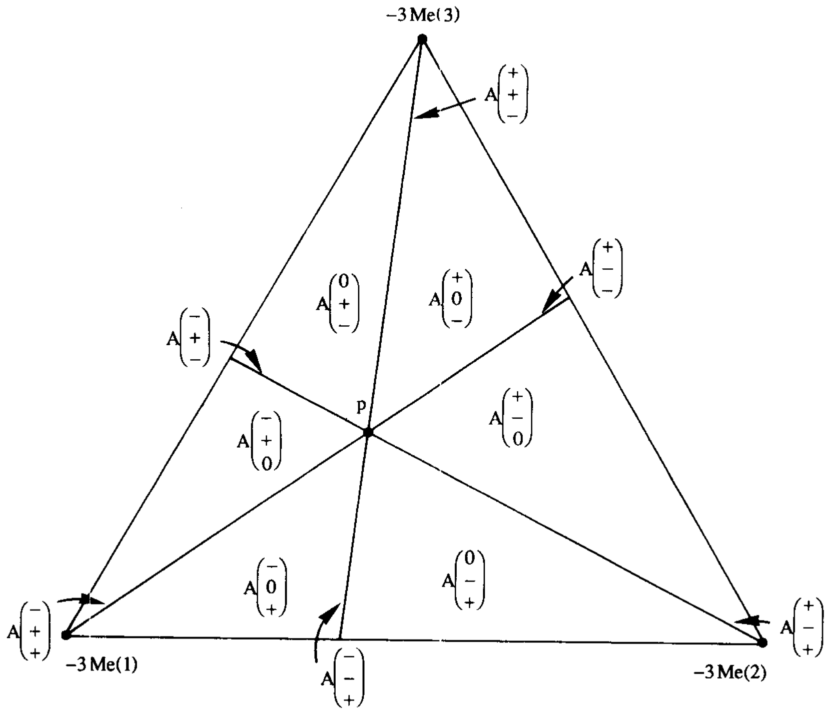

Figure 3.1

The starting point p of the algorithm is an end point of a uniquely determined 1-dimensional simplex $\sigma(p, p^1)$ in $A(s^0)$ and therefore $\sigma(p, p^1)$ is also s^0-complete. Let $a(S^1)$ be the vector label of p^1 then the algorithm is initiated by making a linear programming pivot step with $(a(S^1)^T, 1)^T$ in the system

$$\lambda \begin{bmatrix} a(S^0) \\ 1 \end{bmatrix} - \sum_{h=1}^{n} \mu_h s_h^0 \begin{bmatrix} e(h) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{3.2}$$

If by this pivot step λ becomes first 0, the algorithm moves to the 1-simplex $\sigma(p^1, p^2)$ in $A(s^0)$ adjacent to $\sigma(p, p^1)$ and continues with making a pivot step with $(a(S^2)^T, 1)^T$, where

$a(S^2)$ is the vector label of p^2. Otherwise, one of the μ_h's must become first 0. Suppose that μ_k becomes zero first. Then the algorithm terminates in case there are only two players. If there are more than two players, the algorithm continues with the 2-dimensional simplex $\sigma(p, p^1, p^2)$ in $A(\overline{s})$ containing $\sigma(p, p^1)$ as a facet, where $\overline{s}_k = 0$ and $\overline{s}_h = s_h^0$ for $h \neq k$.

In general the algorithm generates, for varying feasible sign vectors s, a sequence of adjacent t-dimensional simplices in $A(s)$, having s-complete common facets. In each simplex $\sigma(w^1, \cdots, w^{t+1})$ a linear programming pivot step is made with one of the variables in (3.1) in order to determine which other variable becomes first 0. To prevent degeneracy we perturb the right hand side of (3.1). If for some $j \in \{1, \cdots, t+1\}, \lambda_j$ becomes 0, then the facet τ opposite to w^j of σ is also s-complete. If this facet does not lie in the boundary of $A(s)$, there is exactly one t-simplex $\overline{\sigma}$ in $A(s)$ having τ also as a facet. Let $\overline{w}$ be the vertex of $\overline{\sigma}$ opposite to τ, then the algorithm continues by making a pivot step in (3.1) with $(a(\overline{S})^T, 1)^T$, where $a(\overline{S})$ is the vector label of $\overline{w}$. If τ lies in the boundary of $A(s)$ then either τ is a $(t-1)$-simplex in $A(\overline{s})$ with $|I^0(\overline{s})| = |I^0(s)| - 1$ or τ lies in the boundary of U.

Lemma 3.1: An s-complete facet in $A(s)$ does not lie in the boundary of U.

Proof: Suppose that τ is an s-complete $(t-1)$-simplex in $A(s)$, lying in the boundary of U. Clearly, $x_i = 0$ for all $x \in \tau$ and all i for which $s_i = +1$. Let $y^1, \cdots, y^t$ be the vertices of τ. Therefore $y_i^j = 0$ for all i for which $s_i = +1$. Let $a(S^j)$ be the vector label of vertex y^j, i.e., y^j lies in $C_{S^j}, j = 1, \cdots, t$. According to Lemma 2.2, we must have $i \notin S^j, j = 1, \cdots, t$, for all i for which $s_i = +1$. On the other hand, τ is s-complete. Therefore

$$\sum_{j=1}^{t} \lambda_j \begin{bmatrix} a(S^j) \\ 1 \end{bmatrix} - \sum_{s_h \neq 0} \mu_h s_h \begin{bmatrix} e(h) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{3.3}$$

has a nonnegative solution $\lambda_j^*, j = 1, \cdots, t, \mu_h^*$ for $h \notin I^0(s)$. For all i with $s_i = +1$, since $i \notin S^j$, we have that $a_i(S^j) = -|S^j|/n, j = 1, \cdots, t$. Consequently, for i with $s_i = +1$, the i-th equation at the solution of (3.3) is equal to

$$-\sum_{j=1}^{t} \lambda_j^* |S^j|/n - \mu_i^* = 0.$$

Since $\sum_{j=1}^{t} \lambda_j^* = 1$, at least one of the λ_j^*'s is positive and hence for all i for which $s_i = +1$ we obtain $\mu_i^* < 0$, which contradicts $\mu_i^* \geq 0$.

Q.E.D.

An s-complete facet of a t-simplex in $A(s)$, lying in the boundary of $A(s)$, must therefore be a $(t-1)$-simplex in $A(\overline{s})$ with $\overline{s}_j \neq 0$ for some $j \in I^0(s)$ and $\overline{s}_h = s_h$ for all $h \neq j$. Then the algorithm continues with making a pivot step with $\overline{s}_j(e(j)^T, 0)^T$.

Finally, by making a pivot step in (3.1) for a t-simplex σ in $A(s)$, one of the μ_h's may become first 0. Because of the perturbation of the right hand side we may assume that only one of the μ_h's, say μ_k, becomes 0. If s_k is not the only positive or negative component of s, then τ is a facet of just one $(t+1)$-simplex $\overline{\sigma}$ in $A(\overline{s})$, where $\overline{s}_k = 0$ and $\overline{s}_h = s_h$ for $h \neq k$. Let $\overline{w}$ be the vertex of $\overline{\sigma}$ opposite to σ and let $a(\overline{S})$ be the vector label of $\overline{w}$, then the algorithm continues by making a pivot step with $(a(\overline{S})^T, 1)^T$. Suppose now that s_k is the only positive or the only negative component of s, then system (3.1) implies that when we disregard the perturbation and add up the first n equations that all μ_h's must be zero. Therefore the system

$$\sum_{j=1}^{t+1} \lambda_j \begin{bmatrix} a(S^j) \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

has a nonnegative solution $\lambda_j^*, j = 1, \cdots, t+1$. For $j = 1, \cdots, t+1$, let δ_j^* be defined by

$$\delta_j^* = \lambda_j^* / (\sum_{i=1}^{t+1} \lambda_i^* |S^i|/n),$$

then we get

$$\sum_{i \in S^j} \delta_j^* = 1 \text{ for } i = 1, \cdots, n.$$

Hence, the subset $B^* = \{S^1, \cdots, S^{t+1}\}$ is balanced. We remark that some of the λ_j^*'s and therefore some of the δ_j^*'s might be equal to zero. In that case we restrict ourselves to the balanced subset of coalitions S^j for which $\lambda_j^* > 0$. The point $u^* = \sum_{j=1}^{t+1} \lambda_j^* w^j$ can be considered to approximately lie in $\bigcap_{S \in B^*} C_S$ in the sense that u^* lies close to a point in C_S for any $S \in B^*$. Hence, the point $u^* + \tau(u^*)e$ can be taken as an approximating core element.

For $u \in U$, let $\tau^N(u)$ be defined by

$$\tau^N(u) = \max\{r \in R \mid u + re \in v(N)\}.$$

As a measure of accuracy of approximation at u^* one could consider the nonnegative number $\tau(u^*) - \tau^N(u^*)$. If the latter number is too large one may restart the algorithm with a simplicial subdivision of U having a smaller mesh size and with p equal to u^*. Now, let $(G^1, G^2, \cdots)$ be a sequence of triangulations of U with mesh size tending to zero and let $u^{k^*} + \tau(u^{k^*})e$ be the approximating core element found with the algorithm applied for the triangulation $G^k, k = 1, 2, \cdots$. Let B^{k^*} be the set of balanced coalitions corresponding to the vertices of the final simplex σ^k containing u^{k^*}, for all k. Then there exists a subsequence $k_1, k_2, \cdots$, such that $B^{k_j^*} = B^*$ for some balanced set B^* and $u^{k_j^*}$ converges to some u^* in U. Since the vertices of σ^k on this subsequence also converge to u^* and each C_S is closed, we obtain that $u^* \in \bigcap_{S \in B^*} C_S$ and hence that $u^* + \tau(u^*)e$ lies in the core, due to the balancedness of B^*. Notice that $\tau(u^*) - \tau^N(u^*)$ must be zero.

Because the number of simplices of any triangulation G^k in the sequence is finite and due to the perturbation to avoid degeneracy, the algorithm cannot visit a simplex more than once and therefore finds for each k within a finite number of iterations an approximating core element. Moreover, within a finite number of restarts, any accuracy of approximation will be reached.

4. Concluding remarks

In this paper we presented an algorithm for computing an approximating core element of a balanced game. The algorithm can be considered as an adjustment process. During the process, the payoffs for the players are adjusted simultaneously, in order to make every player balanced. At the starting point players in the coalition S^0 can be considered to be overbalanced whereas the players outside S^0 are underbalanced. Overbalanced players are given more payoff and underbalanced ones less payoff. As soon as a new coalition is added to the current one, one player becomes balanced and he is kept balanced. In general, when the algorithm operates in $A(s)$ at the vector $x = \sum_{j=1}^{g+1} \lambda_j y^j$ obtained from the solution of system (3.1), we call a player h overbalanced if $s_h = +1$, underbalanced if $s_h = -1$, and balanced if $s_h = 0$. During the process the payoffs of the overbalanced (underbalanced) players are in principle increased (decreased) in order to make them (more) balanced, whereas the payoffs of the balanced players are adjusted to keep them balanced. When a new coalition is added to the set of current coalitions, a not-balanced player becomes balanced. Sometimes a coalition is deleted from the current set, making a balanced player not balanced anymore. This property guarantees that the process will terminate with a set of coalitions such that all players are balanced and hence an approximating core element has been found.

When we apply the algorithm to market games or exchange markets, the adjustment of the corresponding process seems to be very natural and intuitive in terms of economics.

References

[1] R. G. Aumann: The core of a cooperative game without side payments, *Trans. Amer. Math. Soc.*, Vol. 98 (1961), 539–552.

[2] R. G. Aumann and B. Peleg: Von Neumann-Morgenstern solutions to cooperative games without side payments, *Bull. Amer. Math. Soc.*, Vol. 66 (1960), 173–179.

[3] C. Berge: *Topological Spaces*, Oliver and Boyd, Edinburgh, U.K., 1963.

[4] O. N. Bondareva: The core of an n person game, *Vestnik Leningrad University Mathematics*, Vol. 13 (1962), 141–142.

[5] T. M. Doup, G. van der Laan and A. J. J. Talman: The $(2^{n+1}-2)$-ray algorithm: A new simplicial algorithm to compute economic equilibria, *Mathematical Programming*, Vol. 39 (1987), 241–252.

[6] C. E. Lemke and G. T. Howson: Equilibrium points of bi-matrix games, *SIAM Journal of Applied Mathematics*, Vol. 12 (1964), 413–423.

[7] H. E. Scarf: The core of an N person game, *Econometrica*, Vol. 35 (1967), 52–69.

[8] L. S. Shapley: On balanced sets and cores, *Naval Res. Rogist. Quart.*, Vol. 14 (1967), 453–460.

[9] L. S. Shapley: On balanced games without side payments, in T. C. Hu and S. M. Robinson eds., *Mathematical Programming*, Academic Press, New York, 1973, 261–290.

Kazuya Kamiya
Faculty of Economics
Osaka University
Machikaneyama
Toyonaka, Osaka 560
Japan

Dolf Talman
Dept. of Econometrics
Tilburg University
P. O. Box 90153
5000 LE Tilburg
The Netherlands