RESEARCH PAPER NO. 1967

A Decomposition Algorithm for N-Player Games

Srihari Govindan
Robert Wilson

August 2007

This work was funded in part by a grant from the National Science Foundation.

ABSTRACT

An N-player game can be approximated by adding a coordinator who interacts bilaterally with each player. The coordinator proposes strategies to the players, and his payoff is maximized when each player's optimal reply agrees with his proposal. When the feasible set of proposals is finite, a solution of an associated linear complementarity problem yields an approximate equilibrium of the original game. Computational efficiency is improved by using the vertices of Kuhn's triangulation of the players' strategy space for the coordinator's pure strategies. Computational experience is reported.

A DECOMPOSITION ALGORITHM FOR N-PLAYER GAMES

SRIHARI GOVINDAN AND ROBERT WILSON

Abstract. An N-player game can be approximated by adding a coordinator who interacts bilaterally with each player. The coordinator proposes strategies to the players, and his payoff is maximized when each player's optimal reply agrees with his proposal. When the feasible set of proposals is finite, a solution of an associated linear complementarity problem yields an approximate equilibrium of the original game. Computational efficiency is improved by using the vertices of Kuhn's triangulation of the players' strategy space for the coordinator's pure strategies. Computational experience is reported.

1. Introduction

A hallmark of economic models of production and exchange is the role of price-mediated markets to decentralize the process of arriving at an allocation. In a standard Walrasian formulation one seeks an equilibrium price vector, i.e. one that clears all markets. In the *tatonnement* process hypothesized by Walras [12], an auctioneer proposes a price vector, then agents respond individually with their demands and supplies, and then the auctioneer adjusts prices appropriately. This iterative process continues until aggregate demand equals aggregate supply for all commodities.

A key element of this formulation is the addition of the auctioneer. The substance of an iteration is (1) the auctioneer's announcement of a public signal (proposed prices), (2) responses by the agents individually to the public signal, and (3) the auctioneer's revision of the public signal. The process is simplified by the facts that the public signal is sufficient for each agent to determine an optimal reply, and that market clearing is all that matters to the auctioneer.[1]

The analog for an N-player game adds a coordinator as an extra agent who interacts bilaterally with each player. In each iteration the coordinator proposes a profile of strategies

Date: July 5, 2005; revised August 1, 2007.

Key words and phrases. game theory, computation, equilibrium, decomposition
JEL Classification: C63.

This work was funded in part by a grant from the National Science Foundation. We are grateful for suggestions from a referee.

[1]A practical version of *tatonnement* is used in linear programming: in the Dantzig-Wolfe [2] decomposition algorithm, from a master problem one obtains transfer prices (dual variables on system constraints) that are sent to the subproblems, each of which responds with an optimal vertex of its own feasible set when it must pay these prices for system resources.

for the players. Each player responds to the proposed profile with his strategy that is an optimal reply if all other players use their proposed strategies. If some player's response differs from the strategy proposed then the coordinator adjusts her proposals. The process ends when the proposed profile agrees with the profile of players' optimal replies. The end result is a Nash equilibrium of the original game.

In this article we present an algorithm that is based on a similar construction. The mathematical formulation states the conditions for a Nash equilibrium of the expanded game that includes the coordinator as an extra agent. The algorithm proceeds by adjusting the coordinator's proposal until the players' responses agree with the strategies proposed. Thus the expanded game is an 'imitation game' in which the goal of one player, the coordinator, is to propose the same strategies as the other players adopt in reply. A similar construction is used by McLennan and Tourky (unpublished) in their method for approximating a fixed point of an upper-semi-continuous convex-valued correspondence by an equilibrium of a 2-player game.

The formulation is established in §2. In §3 we characterize an equilibrium of a finite approximation of the game as a solution of a linear complementarity problem. In principle this could be solved using the Lemke-Howson [8] algorithm but in §4 we present a more efficient algorithm obtained when the coordinator's feasible proposals are vertices of Kuhn's [7] triangulation of the strategy space described in Appendix A. Computational experience is reported in §5 and Appendix C.

2. Formulation

The N-player game G has players indexed by n in $\mathcal{N} = \{1 \ldots, N\}$. The game is specified by each player n's sets S_n and $\Sigma_n = \Delta(S_n)$ of pure and mixed strategies, and by his payoff function $G_n : \Sigma \to \mathbb{R}^{S_n}$, where $\Sigma = \prod_{m \in \mathcal{N}} \Sigma_m$. This payoff function is multilinear in other players' mixed strategies and independent of n's strategy, and thus characterized by its values for profiles of pure strategies, i.e. the vertices of Σ. For each proposed profile $\tau \in \Sigma$ of mixed strategies, $G_{ns}(\tau)$ is n's payoff from his pure strategy $s \in S_n$ if each other player $m \neq n$ uses the mixed strategy $\tau_m \in \Sigma_m$. A profile τ is a Nash equilibrium of G if for each player n the proposed strategy τ_n is an optimal reply; that is, $\tau_n \in \arg\max_{\sigma_n \in \Sigma_n} \sum_{s \in S_n} \sigma_{ns} G_{ns}(\tau)$.

The expanded game G° includes the coordinator as an extra player. In this game the coordinator's set of pure strategies includes all profiles in Σ. In G° a profile (σ, τ) specifies the profile $\sigma = (\sigma_1, \ldots, \sigma_N)$ of players' replies to the coordinator's proposal $\tau = (\tau_1, \ldots, \tau_N)$. Player n's payoff is $G^\circ_{ns}(\sigma, \tau) = G_{ns}(\tau)$ from his pure strategy s, as in game G if τ accurately predicts other players' strategies. The coordinator's payoff from her pure strategy τ is

$G^\circ_{0\tau}(\sigma,\tau) = \sum_n u_n(\sigma_n,\tau_n)$, where each function $u_n : \Sigma_n \times \Sigma_n \to \mathbb{R}$ is such that, given σ_n, her payoff $u_n(\sigma_n,\tau_n)$ is maximized uniquely by choosing $\tau_n = \sigma_n$. Obviously:

Theorem 2.1. (σ,τ) is an equilibrium of G° iff $\sigma = \tau$ and σ is an equilibrium of G.

This is the *decomposition principle*. The computational advantage of decomposition is that it replaces the multilateral interactions among the N players with N bilateral interactions, one between each player and the coordinator. Thus, the expanded game is a collection of 2-player games connected solely by the coordinator's role in each.

An implementation of the decomposition principle in a practical algorithm restricts the coordinator to mixtures over a *finite* set of feasible proposals. In this case an equilibrium of the expanded game provides only an approximate equilibrium of the original game.

The following lemma is the basis for Theorem 2.3 below.

Lemma 2.2. Let $E(G)$ be the set of Nash equilibria of the game G. For each $\varepsilon > 0$ there exists $\delta > 0$ with the following property. A strategy profile σ is within ε of $E(G)$ if for each n there exists $\hat{g}_n \in \mathbb{R}^{S_n}$ such that $\|\hat{g}_n - G_n(\sigma)\| \leqslant \delta$ and $\sigma_n \in \arg\max_{\sigma'_n \in \Sigma_n} \sigma'_n \cdot \hat{g}_n$.

Proof. Given $\varepsilon > 0$, let F be the compact set of points in Σ at distances at least ε from $E(G)$. We show that there exists $\delta > 0$ such that the condition of the lemma is violated at every point in F. Define $\alpha : F \to \mathbb{R}$ via $\alpha(\sigma) = \max_{n,\sigma'_n}(\sigma'_n - \sigma_n) \cdot G_n(\sigma)$. Because α is continuous and strictly positive on F, there exists $\delta > 0$ such that $\alpha(\sigma) > 2\delta$ for all $\sigma \in F$. That is, for each $\sigma \in F$ there exists some n and $\sigma'_n \in \Sigma_n$ such that $(\sigma'_n - \sigma_n) \cdot G_n(\sigma) > 2\delta$. Further, if $\hat{g}_n \in \mathbb{R}^{S_n}$ and $\|\hat{g}_n - G_n(\sigma)\| \leqslant \delta$ then:

$$(\sigma'_n - \sigma_n) \cdot \hat{g}_n = (\sigma'_n - \sigma_n) \cdot (\hat{g}_n - G_n(\sigma) + G_n(\sigma)) \tag{1}$$

$$= (\sigma'_n - \sigma_n) \cdot (\hat{g}_n - G_n(\sigma)) + (\sigma'_n - \sigma_n) \cdot G_n(\sigma) \tag{2}$$

$$> (-\delta) + (2\delta) = \delta > 0\,. \tag{3}$$

Hence, $\sigma'_n \cdot \hat{g}_n > \sigma_n \cdot \hat{g}_n$, which implies that $\sigma_n \notin \arg\max_{\sigma'_n \in \Sigma_n} \sigma'_n \cdot \hat{g}_n$. □

To use this result we construct a game $\tilde{G}$ that approximates the expanded game G°. From an equilibrium of the approximate game $\tilde{G}$ we obtain an approximate equilibrium of G° that yields an approximate equilibrium of G.

To define the game $\tilde{G}$, we need a few definitions. For each n, let $f_n : \mathbb{R}^{S_n} \to \mathbb{R}$ be a strictly convex and differentiable function. For each $\tau_n \in \Sigma_n$, let $T_n[\tau_n]$ be its tangent function at τ_n evaluated on Σ_n; that is, $T_n[\tau_n](\sigma_n) = a_n(\tau_n) \cdot \sigma_n$ for each $\sigma_n \in \Sigma_n$, where

$$a_n(\tau_n) = [f_n(\tau_n) - \tau_n \cdot \nabla f_n(\tau_n)]\mathbf{1} + \nabla f_n(\tau_n) \tag{4}$$

and $\mathbf{1}$ is an $|S_n|$-dimensional vector of ones.

On Σ define $f = \sum_n f_n$ and for each $\tau \in \Sigma$ define $T[\tau] = \sum_n T_n[\tau_n]$ by $f(\sigma) = \sum_n f_n(\sigma_n)$ and $T[\tau](\sigma) = \sum_n T_n[\tau_n](\sigma_n)$. By our assumptions on the f_n's there exists a constant $C > 0$ such that for all $\tau \neq \sigma$,

$$0 < f(\sigma) - T[\tau](\sigma) \leqslant C\|\tau - \sigma\| \,. \tag{5}$$

Fix $\varepsilon > 0$ and let $\delta > 0$ be as in Lemma 2.2. Choose $\eta > 0$ sufficiently small that if $\|\sigma - \sigma'\| \leqslant \eta$ then $(\forall n)$ $\|G_n(\sigma) - G_n(\sigma')\| \leqslant \delta$. The strict inequality above implies that there also exists a constant $c > 0$ such that $c < f(\sigma) - T[\tau](\sigma)$ for all $\tau, \sigma \in \Sigma$ with $\|\sigma - \tau\| > \eta$. Now let P be a finite set of points in Σ such that every point in Σ is within c/C of some point in P.

In the approximate game $\tilde{G}$, the coordinator's set of pure strategies is P, and for each player n his set of pure strategies is S_n, the same as in game G. Player n's payoff from his pure strategy $s \in S_n$ is $\tilde{G}_{ns}(\sigma, \tau) = \sum_{p \in P} G_{ns}(p)\tau_p$ when the coordinator's mixed strategy is $\tau \in \Delta(P)$, independently of players' actual mixed strategies in σ. The coordinator's payoff from her pure strategy $p \in P$ is $\tilde{G}_{0p}(\sigma, \tau) = T[p](\sigma)$, where 0 indicates the coordinator.

In the notation before Theorem 2.1 this corresponds to specifying $u_n(\sigma_n, \tau_n) = T_n[\tau_n](\sigma_n)$ to construct the coordinator's payoffs in G°, but in the approximate game $\tilde{G}$ the coordinator's pure strategies are limited to points in the finite set P, and her mixed strategy τ is a randomization over points in P rather than a point in Σ. This accounts for the differences between equilibria of G° and $\tilde{G}$. If τ is a nontrivial randomization then the coordinator proposes a correlated equilibrium but players' responses are uncorrelated. In effect, the coordinator proposes a correlated profile of players' mixed strategies in P that approximates the players' profiles of best responses when they are not confined to P, and which are uncorrelated except for their mutual dependence on the coordinator's proposal.

Our main theorem states that an equilibrium of the approximate game $\tilde{G}$ yields an approximate equilibrium of G° and hence also of G.

Theorem 2.3. If (σ, τ) is an equilibrium of $\tilde{G}$ then σ is within ε of $E(G)$.

Proof. Suppose $(\sigma, \tau) \in \Sigma \times \Delta(P)$ is a mixed-strategy equilibrium of $\tilde{G}$. Then for each player n his mixed strategy σ_n is an optimal reply to $\hat{g}_n \equiv \sum_p G_n(p)\tau_p$. That is, $\sigma_n \in \arg\max_{\sigma'_n \in \Sigma_n} \sigma'_n \cdot \hat{g}_n$. For the coordinator, we claim that each pure strategy in P that is optimal against $\sigma \in \Sigma$ is within η of σ. Indeed, there exists $p \in P$ such that $\|\sigma - p\| \leqslant c/C$, so for any p' such that $\|\sigma - p'\| > \eta$,

$$\tilde{G}_{0p'}(\sigma, \tau) = T[p'](\sigma) < f(\sigma) - c \leqslant T[p](\sigma) + C\|\sigma - p\| - c \leqslant T[p](\sigma) = \tilde{G}_{0p}(\sigma, \tau), \tag{6}$$

which verifies that for the coordinator p is a better reply than p' against σ. Thus all the pure best replies for the coordinator are within η of σ. Consequently, if $\tau_p > 0$ then $(\forall\, n)\ \|G_n(p) - G_n(\sigma)\| \leqslant \delta$. It then follows that $(\forall\, n)\ \|\hat{g}_n - G_n(\sigma)\| \leqslant \delta$. Hence Lemma 2.2 implies that σ is within ε of $E(G)$. □

The only requirements imposed on the f_n's were convexity and differentiability. Therefore, even if we are given a pair (δ, η) as above, in the proof of Theorem 2.3 f had to be approximated at quite a few points; viz., every point in Σ had to be within c/C of some point in P, not just within η of some point in P. In Section 4 we overcome this deficiency by using a specific functional form for f for which it suffices that the distance between any point in Σ and some point in P is of order η rather than c/C.

Even with the function f we use, as a practical matter it is difficult to establish the values of δ and η required to ensure that (σ, τ) is an equilibrium of $\tilde{G}$ only if σ is ε-close to an equilibrium of G. The algorithm presented below therefore takes η as the fundamental measure of accuracy, i.e. the closeness of the approximation is measured by the closeness of the points in P that are the pure strategies of the coordinator.

3. Formulation as a Linear Complementarity Problem

In this section the conditions for an equilibrium of $\tilde{G}$ are displayed as a linear complementarity problem (Eaves [3]). This problem can be solved by the Lemke-Howson algorithm (von Stengel [9, 10]), but in §4 we present a variant that is more efficient than the usual versions of the Lemke-Howson algorithm.

The standard form of a linear complementarity problem seeks a solution (x, y) to:

$$Mx + y = q\,, \quad x \geqslant 0\,, \quad y \geqslant 0\,, \quad x \perp y\,, \tag{7}$$

where M is a square $m \times m$ matrix and q is an m-vector. Here we specify the matrix M by describing the conditions for an equilibrium of the approximate game $\tilde{G}$. The vector q is all zeroes except for the sum constraints that require probabilities to add to 1. The variables used are partitioned as:

$$x = (\sigma, \tau, v, V) \qquad \text{and} \qquad y = (t, u, w, W)\,. \tag{8}$$

Their specifications are shown in Table 1.

We assume a version of the Lemke-Howson algorithm in which the computations are parameterized by λg where $\lambda \in \mathbb{R}$ and the vector $g = (g_{ns}) \in \mathbb{R}^{\sum_n |S_n|}$ is generic and consists of a bonus for each pure strategy s of each player n. The game with the bonuses is denoted by $\tilde{G} \oplus \lambda g$. The algorithm starts with an equilibrium of $\tilde{G} \oplus \lambda g$ at $\lambda = \infty$. Then the parameter λ declines (not necessarily monotonically) until $\lambda = 0$, which yields an equilibrium of $\tilde{G}$. By

Table 1. List of Variables

	Mixed Strategies
σ_{ns}	$= \Pr\{n$ chooses his strategy $s \in S_n\}$
τ_p	$= \Pr\{0$ chooses her strategy $p \in P\}$
	Optimality Slacks
t_{ns}	$= n$'s slack variable for his strategy $s \in S_n$
u_p	$=$ 0's slack variable for her strategy $p \in P$
	Payoffs
v_n	$= n$'s payoff from his interaction with 0
V	$=$ 0's payoff from her interaction with all players
	Probability Slacks
w_n	$=$ slack variable in n's sum constraint
W	$=$ slack variable in 0's sum constraint

allowing negative values of λ one can continue to find additional equilibria each time that λ crosses zero.[2]

The conditions for an equilibrium of $\tilde{G} \oplus \lambda g$ are:

$$\sum_{p \in P} G_{ns}(p)\tau_p + g_{ns}\lambda - v_n + t_{ns} = 0 \qquad (\forall\, n \in \mathcal{N}, s \in S_n) \tag{9}$$

$$\sum_{p \in P} \tau_p + W = 1 \tag{10}$$

$$\sum_{n \in \mathcal{N}} \sum_{s \in S_n} a_{ns}(p_n)\sigma_{ns} - V + u_p = 0 \qquad (\forall\, p \in P) \tag{11}$$

$$\sum_{s \in S_n} \sigma_{ns} + w_n = 1 \qquad (\forall\, n \in \mathcal{N}) \tag{12}$$

For the algorithm described in §4 it is convenient to formulate the coordinator's interaction with each player separately. That is, the coordinator's strategies are represented in behavioral form. This entails two minor modifications. Modify Table 1 by letting u_{np_n} be 0's slack variable for her behavioral strategy $p_n \in P_n$, and by letting V_n be 0's payoff from her interaction with player n; i.e. $V = \sum_n V_n$ and $u_p = \sum_n u_{np_n}$. Then the equilibrium condition (3) is altered to:

[2]For details of these aspects see [1, 4, 5, 6]. The payoff and slack variables v_n, V, w_n, W are used only in an initial step of the algorithm that makes each v_n and V basic and w_n and W nonbasic, after which they remain so throughout the algorithm.

(13)
$$\sum_{s\in S_n} a_{ns}(p_n)\sigma_{ns} - V_n + u_{np_n} = 0 \qquad (\forall\, n \in \mathcal{N}, p_n \in P_n)\,,$$

and the standard complementary condition ($\tau_p > 0$ only if $u_p = 0$) is replaced by: $\tau_p > 0$ only if $(\forall\, n \in \mathcal{N})$ $u_{np_n} = 0$. This modification enables the computations to be organized into $1+N$ separate tableaus of detached coefficients for the equations they represent. Tableau $\mathcal{T}_0$ represents equations (1) and (2), and for each player n the tableau $\mathcal{T}_n$ represents equations (5) and (4) for the given value of n.

4. A Modified Lemke-Howson Algorithm

In this section we modify the Lemke-Howson algorithm to improve its computational efficiency. These alterations depend on a particular specification of the set P of proposals available to the coordinator and a particular function f.

For each player n the set P_n of the coordinator's feasible proposals is the set of rational points in Σ_n for which the divisor is an integer K_n. Thus in Theorem 2.3 the accuracy parameter is $\eta = \max_n 1/K_n$. In practice this is implemented by representing the simplex Σ_n of player n's mixed strategies as the set of nonnegative points summing to the integer K_n. With this convention, the coordinator's feasible set P_n of proposals to player n comprises all the integer points in Σ_n, and similarly, the proposals in P are the integer points in Σ. Let $\mathcal{P}$ be the triangulation of Σ described by Kuhn [7] that has P as its vertex set. Kuhn's triangulation is described in Appendix A.

For each n, the function f_n we use is $f_n(\sigma_n) = \frac{1}{2}\sum_{s_n\in S_n} \sigma^2_{n,s_n}$. An important advantage of combining this function with Kuhn's triangulation is that it is sufficient to restrict the coordinator's strategies to mixtures whose support is the vertices of either a principal simplex of some face of Σ or a maximal proper face of it. This has the advantage that one never needs to compute the entire set P nor the triangulation $\mathcal{P}$: from the (principal) simplex of the triangulation currently used by the algorithm, one obtains the next one needed by the algorithm by invoking the simple rules in Appendix A. Moreover, the players' payoffs are computed only at those points in P actually encountered along the path of the algorithm. Indeed, only the data for the currently active simplex needs to be retained in the tableaus.[3]

[3]Our implementation computes and adjoins to the tableau only when needed by the algorithm the column $G(p)$ of players' pure-strategy payoffs from the proposal p, and the row $a_n(p_n)$ of the coordinator's payoffs from her interaction with player n. We also delete a column or row from the tableau immediately after it becomes unused. This reduces the number of multiplications involved in each pivoting operation. To further minimize multiplications we compute payoffs via a recursive function that minimizes redundant multiplications. Incidentally, these multiplications are substantially faster when proposals are integers.

The algorithm consists of a finite sequence of pivoting operations (Gaussian eliminations). Each pivoting operation switches the status of a variable that is currently nonbasic (i.e. constrained to zero) and a variable that is basic, i.e. the nonbasic variable is increased from zero until the nonbasic variable becomes zero. In the corresponding tableau these are identified with the pivot column and the pivot row. The algorithm starts with the unique proposal p^* that is the vertex of Σ that is an equilibrium of $\tilde{G} \oplus \lambda g$ when λ is sufficiently large. The initial steps of the algorithm make each v_n and V_n basic and w_n and W nonbasic, after which they remain so throughout the algorithm, and (if p^* is not an equilibrium for $\lambda = 0$) makes λ basic and t_{ns} nonbasic for that player n and his strategy s that first becomes optimal as λ decreases. Thereafter, the sequence of pivots traces a path in the graph of equilibria of $\tilde{G} \oplus \lambda g$ until λ becomes nonbasic. At this point $\lambda = 0$ so one has arrived at an equilibrium of $\tilde{G}$, and thus an approximate equilibrium of G°, for which the players' profile of strategies is an approximate equilibrium of G.

As in the standard Lemke-Howson algorithm, in every iteration the slack variable t_{ns} is nonbasic in $\mathcal{T}_0$ iff the strategy variable σ_{ns} is basic in $\mathcal{T}_n$. The analogous complementarity conditions for the coordinator's strategies are slightly different because we represent them in behavioral form. The strategy variable t_p is basic in tableau $\mathcal{T}_0$ iff in each tableau T_n the slack variable u_{np_n} for the proposal p_n to player n is nonbasic. The variables t_p that are basic in tableau $\mathcal{T}_0$ are always those for which the corresponding proposals p are the vertices of a single simplex of $\mathcal{P}$, called the currently active simplex.

As in the usual Lemke-Howson algorithm, the pivot row in one iteration precisely determines the pivot column in the next iteration. In the modified algorithm the rules for selecting the next pivot column derive from the rules in Appendix A for moving from one simplex of the triangulation to an adjacent one. The rules listed below are stated for the case that (G, g) is generic, as is assumed in Theorem 4.1 presented below.

- If the pivot row was t_{ns} in $\mathcal{T}_0$ then the next pivot column is σ_{ns} in $\mathcal{T}_n$. The next pivot row is found as follows. From the currently active simplex construct the next higher dimensional simplex obtained by adding dimension (n, s). This adds a new proposal p to the active simplex of the coordinator, so the next pivot row is u_{np_n}. In the subsequent iteration the next pivot column is τ_p in $\mathcal{T}_0$.
- If the pivot row was τ_p in $\mathcal{T}_0$ then find the simplex adjacent to the currently active simplex when the proposal p is deleted and a new proposal p' is added, or identify that dropping p yields a lower dimensional simplex on a lower dimensional face of Σ. This yields three cases.

(1) If p makes to each player n a proposal p_n that is also proposed to n by some other vertex of the currently active simplex then the next pivot column is $\tau_{p'}$ in $\mathcal{T}_0$.
(2) If dropping p drops the proposal p_n to some player n [just one player because (G, g) is generic] then the next pivot column is u_{np_n} in $\mathcal{T}_n$ and the next pivot row is $u_{np'_n}$. In the subsequent iteration the next pivot column is $\tau_{p'}$ in $\mathcal{T}_0$.
(3) If dropping p yields a lower dimensional simplex on a lower dimensional face of Σ [just one dimension lower because (G, g) is generic] then some one dimension (n, s) is newly zero on this face. The next pivot column is u_{np_n} in $\mathcal{T}_n$ and the next pivot row is σ_{ns}. In the subsequent iteration the next pivot column is t_{ns} in $\mathcal{T}_0$.

Because these rules determine the next pivot row whenever the pivoting operation occurs in some tableau $\mathcal{T}_n$, it remains only to determine the next pivot row when the pivot column is in $\mathcal{T}_0$. This is done as usual: the next pivot row is the basic variable that first is driven to zero when the pivot column's nonbasic variable is increased from zero.

Theorem 4.1. If (G, g) is generic then the path of the algorithm is a 1-dimensional piecewise-linear manifold whose boundary point at $\lambda = 0$ is an equilibrium of $\tilde{G}$.

Proof. See Appendix B, Theorem B.15. □

Note that this theorem is about the path of the algorithm, not the graph of the equilibria over the half line $0 < \lambda < \infty$. In fact, at every point the algorithm makes an implicit selection among the equilibria in a component whenever there are multiple continuations. This selection stems from the algorithm's reliance on the principal simplices of Kuhn's triangulation.

Figure 1 shows an example of the path of the algorithm, projected onto the simplex Σ_n of one player n's mixed strategies. The figure shows the triangulation $\mathcal{P}_n$ when $m_n = 3$ and $K_n = 5$. The coordinator's set P_n of feasible proposals is the set of vertices of this triangulation. The pivoting operations in tableau $\mathcal{T}_n$ yield the sequence of circled points. At each stage the support in P_n of the coordinator's mixed strategies in $\mathcal{T}_0$ is the set of vertices of the smallest subsimplex that contains the circled point (due to our choice of f). The path starts in the lower right corner and ends in the boundary subsimplex where the path terminates with an arrow.

4.1. **A Slim Version of the Algorithm.** From the above rules for the pivoting operations it is evident that omitting all pivots in tableaus $\mathcal{T}_n$, $n \geqslant 1$, has no effect on the sequence

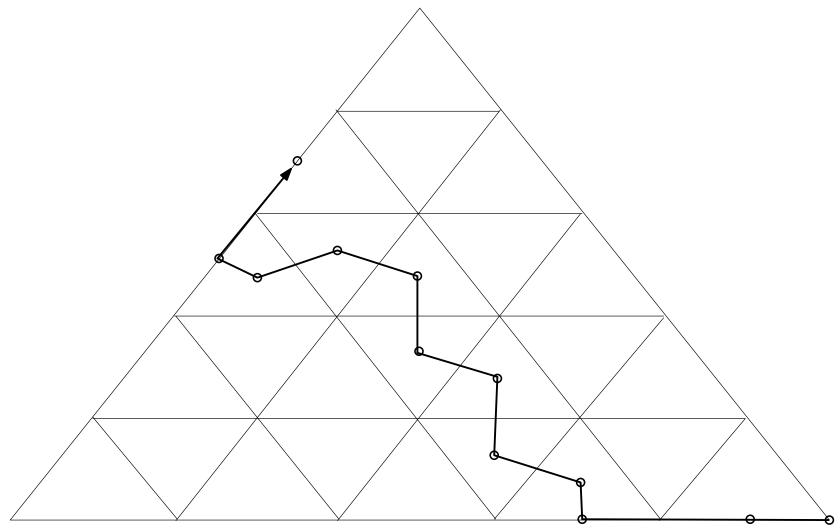

FIGURE 1. Illustrative Path of the Algorithm

of pivots in tableau $\mathcal{T}_0$. At any stage, moreover, from the currently active simplex one can reconstruct the players' strategies that would have been traced by pivoting in their tableaus, as shown in Appendix B, Theorem B.9. Thus, the 'slim' version of the algorithm uses only tableau $\mathcal{T}_0$ and the pivoting operations there. Although the slim version is somewhat faster, all the computational results reported in the next section are based on the full version of the algorithm.

5. COMPUTATIONAL EXPERIENCE

The algorithm is implemented in an experimental code available from the authors and shown here as Appendix D. The code is written in APL, which ordinarily is used only for program development. Each test problem was solved on a laptop computer with a Windows operating system and a CPU speed of 1.83 MHz. The number of CPU cycles was estimated as the product of the elapsed time in seconds and 1.83 MHz; e.g. if the elapsed time was 5000 milliseconds then the estimated CPU time is $5 \times 1.83 = 9.15$ megacycles. Because APL operates as a run-time interpreter it typically uses about three times as many cycles of the CPU as an efficient compiled language such as C or FORTRAN.

For each pair (N, m) where N is the number of players and $m = |S_n|$ is the number of pure strategies for each player n, 100 examples were randomly generated with payoffs from each profile of pure strategies uniformly distributed on $[-1, 1]$. In all examples the bonus vector g had $g_{ns} = 0$ except $g_{n1} = 1$. Results are shown in Appendix C only for pairs (N, m) for which the average elapsed time was less than a few minutes. The same examples were used for each value of $K = 10, 20, 40$, where each $K_n = K = 1/\eta$ and η is the accuracy parameter. The accuracy of payoffs was better than one might infer from the strategy accuracy η. For $K = 40$ and thus $\eta = 0.025$, the average of the maximum difference between the players' equilibrium payoffs in the approximate game $\tilde{G}$ and their actual payoffs in G from those strategies used in the equilibrium of $\tilde{G}$ was 0.005 and in every case this average was in the interval [0.0006, 0.012]. The average of the maximum difference between the expectation (using the coordinator's mixed strategy) of the coordinator's proposals and the players' equilibrium strategies was 0.019.

The test problems verify the importance of Savani and Stengel's [11] demonstration that the Lemke-Howson algorithm's worst-case computing time is exponential in the number of players' strategies. Appendix C includes histograms for some values of (N, m) that suggest that the distribution of computational times has a long tail; indeed, the histograms resemble exponential distributions and in nearly every case the standard deviation of times was as large as the average time. In most cases the average times were strongly affected by a few test problems that required many pivoting operations and payoff evaluations.

References

[1] Blum, B., Shelton, C., Koller, D.: A Continuation Method for Nash Equilibria in Structured Games. *J. Artificial Intelligence Res.* **25**, 457-502 (2006)

[2] Dantzig, G.: *Linear Programming and Extensions*. Princeton: Princeton University Press 1963

[3] Eaves, B.: The Linear Complementarity Problem. *Management Science* **17**, 612-634 (1971)

[4] Govindan, S., Wilson, R.: A Global Newton Method to Compute Nash Equilibria. *J. Econ. Theory* **110**, 65-86 (2002)

[5] Govindan, S., Wilson, R.: Structure Theorems for Game Trees. *Proc. Nat. Acad. Sciences USA* **99**, 9077-9080 (2002)

[6] Govindan, S., Wilson, R.: Computing Nash Equilibria by Iterated Polymatrix Approximation. *J. Econ. Dynamics and Control* **28**, 1229-1241 (2004)

[7] Kuhn, H.: Some Combinatorial Lemmas in Topology. *IBM J. Res. Dev.* **4**, 518-524 (1960)

[8] Lemke, C., Howson, J.: Equilibrium Points of Bimatrix Games. *J. Soc. Industrial and Appl. Math.* **12**, 413-423 (1964)

[9] von Stengel, B.: Computing equilibria for Two-person Games. R. Aumann and S. Hart (eds.), *Handbook of Game Theory*, North-Holland, Amsterdam, **3**, 1723-1759 (2002)

[10] von Stengel, B.: Equilibrium Computation for Two-player Games in Strategic and Extensive Form. N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani (eds.), *Algorithmic Game Theory*, Cambridge Univ. Press, Cambridge UK, Chapter 3, to appear 2007.

[11] Savani, R., von Stengel, B.: Hard-to-Solve Bimatrix Games. *Econometrica* **74**, 397-429 (2006)

[12] Walras, L.: *Elements of Pure Economics*, 1874. London: Allen & Unwin, English translation 1954

Appendix A. Kuhn's Triangulation of the Strategy Space

For each player n, label and order his pure strategies in S_n by successive integers $1, \ldots, |S_n|$. The simplex Σ_n of player n's mixed strategies is assumed to be represented by nonnegative points summing to an integer K_n. With this convention, we require that the coordinator's feasible set P_n of proposals to player n comprises all the integer points in Σ_n, and similarly, the proposals in P are the integer points in Σ.

We now describe our adaptation of Kuhn's [7] triangulation $\mathcal{P}$ of Σ with P as the vertices. For each player n suppose that $T_n \subset S_n$ is an ordered subset of n's pure strategies with elements $i_1 < i_2 < \cdots < i_{m_n}$. We describe simplices of $\mathcal{P}$ in the face of the strategy space $\Sigma = \prod_n \Sigma_n$ spanned by $T = \prod_n T_n$. The description here is confined to the principal simplices, i.e. those with the maximal dimension $\sum_n m_n - N$. We also show the operations used to obtain from a given simplex:

- Each simplex with the same support and the same dimension that is adjacent to the given simplex.
- Each simplex that is a face of the given simplex with one fewer dimension and that belongs to the same face of Σ.
- Each simplex that is a face of the given simplex with one fewer dimension and that belongs to a next lower dimensional face of Σ.
- Each simplex on a next higher dimensional face of Σ for which the given simplex is a face.

These four operations do not exhaust all the possibilities but they include all those encountered on the path of the algorithm when (G, g) is generic, as assumed in Theorem 4.1.

In Kuhn's triangulation, each simplex of $\mathcal{P}$ with support T is characterized by two objects:

- An anchor, which is a vertex with support in T.
- An ordering of all pairs (n, i_j) such that $i_j \in T_n$ and $i_j \neq i_{m_n}$.

The vertices p^k, $k = 0, \ldots, \sum_n m_n - N$ of the simplex are obtained recursively, starting from the anchor p^0. From each p^k one obtains p^{k+1} by adding 1 to the coordinate (n, i_j) that is the k-th in the ordering, and subtracting 1 from coordinate (n, i_{j+1}). In this case (n, i_j) is called the k-th increment even though it entails both an increment and a decrement.

Observe that coordinate (n, i_1) never has a 1 subtracted at any stage, and coordinate (n, i_{m_n}) never has a 1 added to it, while other coordinates are the same at the beginning and end of the sequence of the vertices. For example, if each of 3 players has 4 strategies, each $K_n = 10$, and $T = (2\ 4) \times (1\ 2\ 3) \times (1\ 2\ 3)$ then from the anchor and the order of

k	Player n:	1	2	3	order of increments	
0	$p^0 =$	(0 2 0 8)	(1 2 7 0)	(0 2 8 0)	= anchor	
1	$p^1 =$	(0 2 0 8)	(1 2 7 0)	(1 1 8 0)	$(n, i_j) =$	(3,1)
2	$p^2 =$	(0 3 0 7)	(1 2 7 0)	(1 1 8 0)		(1,2)
3	$p^3 =$	(0 3 0 7)	(1 3 6 0)	(1 1 8 0)		(2,2)
4	$p^4 =$	(0 3 0 7)	(2 2 6 0)	(1 1 8 0)		(2,1)
5	$p^5 =$	(0 3 0 7)	(2 2 6 0)	(1 2 7 0)		(3,2)

TABLE 2. Example of the Vertices of a Simplex of $\mathcal{P}$

increments shown in the right panel of Table 2 one obtains the sequence of vertices shown in the left panel.

Each simplex adjacent to the given simplex, with the same dimension and on the same face of Σ spanned by T, is obtained by one of the following three operations:

(1) If (n, i_j) is first in the ordering then make it last and make p^1 the new anchor.
(2) If (n, i_j) and $(n', i_{j'})$ are two successive increments in the ordering then switch their positions in the ordering.
(3) If (n, i_j) is last in the ordering then make it first. Also, revise the anchor so that the old anchor becomes the new p^1, i.e., from the old anchor p^0 subtract 1 from coordinate (n, i_j) and add 1 to coordinate (n, i_{j+1}) to obtain the new anchor.

The restrictions on these three operations are the following:

(1) Operation (1) is not possible precisely when coordinate (n, i_{j+1}) is actually coordinate (n, i_{m_n}) and it is zero everywhere except in the anchor. Thus dropping the anchor puts one on the face of Σ where coordinate (n, i_{m_n}) is zero.
(2) Operation (2) is not possible precisely when $n = n'$, $j = j' + 1$ and coordinate (n, i_j) is zero in the anchor. In this case one must increment (n, i_j) by 1 before decrementing it; taking out the vertex obtained after incrementing (n, i_j) then puts one on the face of Σ where coordinate (n, i_j) is zero.
(3) Operation (3) is not possible precisely when coordinate (n, i_j) is actually (n, i_1) and it is zero in the anchor. Deleting the last vertex produces a simplex on the face of Σ where coordinate (n, i_1) is zero.

Each of these three restrictions identifies a condition in which a next lower dimensional face of Σ is encountered.

These three operations thus determine all possible adjacent simplices on this face. They also determine the maximal faces of the given simplex that lie on the face of Σ spanned by T. Each maximal face of the given simplex is obtained by deleting one of the increments in the ordering. In particular, those maximal faces of the given simplex on a next lower dimensional face of Σ are obtained iff one of the following holds:

(1) The first pair in the ordering gives a -1 to (n, i_{m_n}) for some n and moves it from 1 to 0. It is impossible to drop the anchor and add a new vertex at the top that performs this increment last.

(2) Player n has two successive increments of the form $(n, i_j), (n, i_{j-1})$ and the increment (n, i_j) moves this coordinate from 0 to 1. It is impossible to reverse this order of increments.

(3) The last increment is (n, i_1) for some n and this increment moves the coordinate from 0 to 1. It is impossible to delete the last vertex and put a new anchor that reverses this order of increments from the old anchor; i.e. subtract 1 from (n, i_1) and add 1 to (n, i_{i_2}) from the old anchor to form a new anchor.

Finally, one can add a new increment (n, i) for which $i \notin T_n$ and thus obtain a next higher dimensional simplex of $\mathcal{P}$ on a next higher dimensional face of Σ.

- If $i < i_1$ then adjoin (n, i) as the last increment in the ordering. Note that i becomes the new value of i_1 for n.
- If $i_j < i < i_{j+1}$ for some $1 \leqslant j < m_n$ and (n, i_j) is the k-th increment then insert (n, i) as the new k-th increment between the old $(k-1)$-th and k-th increments.
- If $i > i_{m_n}$ then adjoin (n, i_{m_n}) as first increment in the ordering and adopt as the new anchor the vertex such that the old anchor becomes second in the sequence of vertices. That is, the new anchor is obtained from the old anchor by subtracting 1 from (n, i_{m_n}) and adding 1 to (n, i). Note that i becomes the new value of m_n.

Appendix B. A Canonical Decomposition for N-Player Games

This Appendix is a self-contained presentation of the theory of the decomposition algorithm.

B.1. **Notation.** As in the text, the set of players is $\mathcal{N} = \{1, \ldots, N\}$. Each player $n \in \mathcal{N}$ has a finite set of pure strategies denoted S_n. His set of mixed strategies is denoted Σ_n. $S \equiv \prod_n S_n$; $\Sigma \equiv \prod_n \Sigma_n$; for each n, $S_{-n} \equiv \prod_{m \neq n} S_m$ and $\Sigma_{-n} \equiv \prod_{m \neq n} \Sigma_{-m}$. Let d be the dimension of Σ. For any nonempty set T_n of S_n for player n, $\Sigma_n(T_n)$ is the set of mixed strategies in Σ_n whose support is contained in T_n. Given a subset T_n for each player, we use T and $\Sigma(T)$ to denote the corresponding product spaces and d^T to denote the dimension of $\Sigma(T)$.

With these player and strategy sets a game G is now specified as a collection of payoff functions, one per player, i.e. a game is specified as a vector G in $\mathcal{G} \equiv \mathbb{R}^{\mathcal{N} \times S}$. Given a game G, we write $G_n(s)$ to denote player n's payoff from the strategy profile $s \in S$.

For ease in exposition, we label the pure strategies of all the players as consecutive positive integers. Specifically, each strategy set S_n is given as $\{ i^*_{n-1} + 1, \ldots, i^*_n \}$ where $i^*_0 = 0$. (Thus player n has $i^*_n - i^*_{n-1}$ pure strategies, and $d = i^*_N - N$). Let $I = \{ 1, \ldots, i^*_N \}$ be the set of all the pure strategies for all players. (Σ is then a subset of $\mathbb{R}^I$.) Let $I^* = \{ i^*_1, \ldots, i^*_N \}$ be the collection of the "last" strategies of the players. For a typical nonempty subset T_n of S_n, we denote the elements as $i^1(T_n) < \cdots < i^*(T_n)$. For $T = \prod_n T_n$, $I(T) \subseteq I$ is the collection of all the strategies in T_n for each n, and $I^*(T)$ is the set of all $i^*(T_n)$.

Let p be a a positive integer that is fixed throughout. For each n, let Λ_n be the set of $\lambda_n \in \Sigma_n$ such that $p\lambda_n$ is a vector with integer coordinates. For each subset T_n, let $\Lambda_n(T_n) = \Lambda_n \cap \Sigma_n(T_n)$. $\Lambda \equiv \prod_n \Lambda_n$; for each n, $\Lambda_{-n} = \prod_{m \neq n} \Lambda_m$; for each subset T of S, $\Lambda \equiv \prod_n \Lambda_n(T_n)$.

Given a game G we construct a new game $\tilde{G}$ that has an extra player, called player 0 or the coordinator. His set of pure strategies is Λ, and his set of mixed strategies is denoted Σ_0. (Σ_0 is viewed as a subset of $\mathbb{R}^\Lambda$.) The payoffs for each player $n \in \mathcal{N}$ in the game $\tilde{G}$ depends only on his strategy and the strategy of player 0, and it is given by $\tilde{G}_n(s_n, \lambda) = G(s_n, \lambda_{-n})$. Player 0's payoff function is defined as follows: if he plays a pure strategy $\lambda = (\lambda_1, \ldots, \lambda_N)$ and the players in $\mathcal{N}$ play the mixed strategy profile σ, his payoff is $\varphi(\lambda, \sigma) = \sum_n \varphi_n(\lambda_n, \sigma_n)$ where $\varphi_n(\lambda_n, \sigma_n) = p\lambda_n \cdot (\sigma_n - .5\lambda_n)$.

For each i, let e_i be the i-th unit vector in $\mathbb{R}^S$; and, for any subset T of S, $e_T \equiv \sum_{i \in T} e_i$. By a slight abuse of notation, if i is a coordinate of player n, we still use e_i to denote the i-th unit vector in $\mathbb{R}^{S_n}$—it should be clear from the context as to what the ambient space is; the same comment applies to vectors of the form e_{T_n}. For $i, j \in S_n$ for some n, let $\delta_{i,j} = p^{-1}(e_i - e_j)$. (The usefulness of the notation lies in the fact that if $\lambda \in \Lambda$ and $\lambda_j > 0$, then $\lambda + \delta_{i,j}$ belongs to Λ as well.) For the special case where $j = i + 1$ (and thus $i \neq i^*_n$), we will denote $\delta_{i,j}$ by ξ_i, i.e. ξ_i is the vector $p^{-1}(e_i - e_{i+1})$.

B.2. **Triangulating the Strategy Space.** We now describe a triangulation of Σ that has Λ as its vertex set. The procedure is derived from Kuhn's [7] procedure for triangulating a cube.

For each $\sigma \in \Sigma$, define $x(\sigma) \in \mathbb{R}^S$ by: for each n and $i \in S_n$, $x_i(\sigma) = p \sum_{i^*_{n-1} < j \leqslant i} \sigma_j$. In other words, $x(\sigma)$ is a representation of the mixed strategy profile σ as a profile of "cumulative distributions" scaled by p. Let $X = \{\, x(\sigma) \mid \sigma \in \Sigma \,\}$; let $V = \{\, x(\lambda) \mid \lambda \in \Lambda \,\}$. X is easily seen to be the set of all $x \in \mathbb{R}^I$ such that $0 \leqslant x_{i^*_{n-1}+1} \leqslant x_{i^*_{n-1}+2} \leqslant \cdots \leqslant x_{i^*_n - 1} \leqslant x_{i^*_n} = p$ for each n. And, V is the set of points in X whose coordinates are all integers. We first describe a triangulation of X with V as the vertex set. The triangulation that it implies for Σ is then derived in a straightforward manner since the map sending σ to $x(\sigma)$ is a linear homeomorphism between Σ and X. The following lemma is the basis for our triangulation.

Lemma B.1. Each $x \in X$ has a unique decomposition of the form $x = \sum_{l=0}^{k} \alpha^l v^l$, where: $\alpha^l > 0$ for all l, and $\sum_l \alpha^l = 1$; $v^l \in V$ for all l, and $v^0 \lneqq v^1 \lneqq \cdots \lneqq v^k \leqslant v^0 + e_{I \setminus I^*}$.

Proof. Given $x \in X$, define $[x]$ to be the integer part of x, i.e., for all coordinates i, $[x]_i$ is the largest integer smaller than x_i. Clearly $[x]$ belongs to V. Let $r = x - [x]$. All the coordinates of r are nonnegative and strictly smaller than 1; also, for all $i \in I^*$, $r_i = 0$, since $x_i = p$. Let $v^0 = [x]$. If $r = 0$, then set $k = 0$ and $\alpha^0 = 1$. Otherwise, we define a finite sequence $a^1, \ldots, a^k$ of positive numbers and a corresponding sequence of pairwise disjoint subsets $I^1, \ldots, I^k$ of subsets of I inductively as follows. Let a^1 be the maximum over i of r_i and let I^1 be the coordinates that achieve this maximum. For $l > 1$, let a^l be the maximum over $\{\, i \notin \cup_{l'<l} I^{l'} \mid r_i \neq 0 \,\}$ of r_i and let I^l be the set of coordinates that achieve this maximum. ($k = l - 1$ for the first integer l for which this maximum does not exist, that is, $r_i = 0$ for all $i \notin \cup_{l'<l} I^{l'}$.) Since $r_i = 0$ for each $i \in I^*$, I^l is contained in $I \setminus I^*$ for each l. For $0 < l \leqslant k$, let $v^l = v^{l-1} + e_{I^l}$; we have that $v^0 \lneqq v^1 \lneqq \cdots \lneqq v^k \leqslant v^0 + e_{I \setminus I^*}$. Let $\alpha^0 = (1 - a^1)$ and, for $0 < l \leqslant k$, let $\alpha^l = (a^l - a^{l+1})$ where a^{k+1} is zero. $0 < \alpha^l < 1$ for all l.

Moreover

$$\sum_l \alpha^l v^1 = (1-a^1)v^0 + (a^1-a^2)(v^0+e_{I^1}) \tag{14}$$

$$+\cdots+(a^{k-1}-a^k)(v^0+e_{I^1}+\cdots+e_{I^{k-1}}) \tag{15}$$

$$+a^k(v^0+e_{I^1}+\cdots+e_{I^k}) \tag{16}$$

$$= v^0+a^1e_{I^1}+\cdots+a^ke_{I^k}=[x]+r=x\,, \tag{17}$$

which thus gives us a decomposition that is obviously unique. □

Remark B.2. It is easy to see that the $k+1$ vertices in the lemma can equivalently be written in the form $v^0, v^0+e_{I^1}, \ldots, v^0+e_{I^k}$ where $\emptyset \neq I^1 \subsetneq \cdots \subsetneq I^k \subseteq I\backslash I^*$.

Define a simplicial complex $\mathcal{D}$ as follows. The vertex set is V. A simplex in $\mathcal{D}$ is the convex hull of a set of vertices of the form $v^0 \leqslant v^1 \leqslant \cdots \leqslant v^k \leqslant v + e_{I\backslash I^*}$. It follows from Lemma B.1 that $\mathcal{D}$ is indeed a simplicial complex that triangulates X.

A simplex of $\mathcal{L}$ is called a principal simplex if it is not a face of another simplex—or equivalently if it has a nonempty interior in X. There exists a simplex characterization of the principal simplices in $\mathcal{L}$, which we now provide.

Lemma B.3. A simplex D of $\mathcal{D}$ is a principal simplex iff there is an ordering (i.e. a bijection) $\pi : \{1,\ldots,d\} \to I\backslash I^*$ such that the vertex set of D is of the form $\{v^0, v^1, \ldots, v^d\}$ where $v^l = v^{l-1} + e_{\pi(l)}$ for $l > 0$.

Proof. As X is d-dimensional, observe first that the principal simplices are those that have $d+1$ vertices. Therefore, any simplex whose vertices are generated by an ordering as in the lemma is clearly a principal simplex. To prove the other way around, suppose that L is a principal simplex with vertex set $v^0 \lneqq v^1 \lneqq \cdots \lneqq v^d$. Since $v^d \leqslant v^0 + e_{I\backslash I^*}$, and $|I\backslash I^*| = d$, the vertices of L satisfy the following properties: (i) for each $0 < l \leqslant d$, we have that $v^l - v^{l-1}$ is unit vector for some coordinate $i \in I\backslash I^*$; (ii) for each $i \in I\backslash I^*$ there exists a unique l such that $v^l - v^{l-1} = e_i$. There is now an implied ordering π and the vertices of L are obtained from v^0 and this ordering as given in the lemma. □

Having triangulated X we now show the properties of the triangulation, call it $\mathcal{L}$, that it implies for Σ. It turns out to be easier to characterize the principal simplices in $\mathcal{L}$ than to characterize any arbitrary simplex of $\mathcal{L}$ directly (which is the main reason for first studying X). The vertices of a principal simplex in $\mathcal{D}$ are obtained successively from some vertex by adding one unit vector at a time in some order. So describing what this operation does in Σ gives us a handle on the principal simplices. Suppose $v = x(\lambda)$. If $v + e_i$ belongs to X, it is obtained from v by adding 1 to coordinate i. This implies that $x^{-1}(v+e_i)$ is obtained from λ by adding p^{-1} to coordinate i and subtracting it from coordinate $i+1$. (Coordinates i and $i+1$ belong to the same player, since if $i = i_n^*$ for some n, then $v+e_i \notin X$.) In other words, $x^{-1}(v+e_i) = \lambda + \xi_i$. We therefore have the following theorem, whose proof is obvious.

Theorem B.4. A simplex L is a principal simplex of $\mathcal{L}$ iff there exists an ordering $\pi : \{1, \dots d\} \to I \backslash I^*$ such its vertex set is of the form $\{\lambda^0, \cdots, \lambda^d\}$ where for each $k > 0$, $\lambda^k = \lambda^{k-1} + \xi_{\pi(k)}$.

Every principal simplex in $\mathcal{L}$ then has a compact representation in the form of a vertex-ordering pair (λ, π).

For each $T \subseteq S$, the triangulation $\mathcal{L}$ induces a triangulation, call it $\mathcal{L}(T)$, of $\Sigma(T)$, which we now study. Again, to do so, we first study the equivalent problem for X. Let $X(T)$ be the set of $x(\sigma)$ such that $\sigma \in \Sigma(T)$. More directly, $X(T)$ is the set of $x \in X$ such that for each n and $i \notin T_n$: $x_i = x_{i-1}$ if $i > i_{n-1}^* + 1$ and zero if $i = i_{n-1}^* + 1$. Let $\mathcal{D}(T)$ be the set of simplices of $\mathcal{D}$ whose vertices belong to $V(T)$. Then $\mathcal{D}(T)$ is a triangulation of $X(T)$.

There exists a simple characterization of the principal simplices of $\mathcal{D}(T)$ analogous to that of the principal simplices of $\mathcal{D}$. To get at this, we need some more notation. For each n and $i^k(T_n)$ define $R(i^k(T_n)) \equiv \{ i \in S_n \mid i^k(T_n) \leqslant i < i^{k+1}(T_n) \}$, where $i^{k+1}(T_n) = i_n^* + 1$ if $k = *$. Observe that for each $x \in X(T)$, $x_i = x_{i^k(T_n)}$ if $i \in R(i^k(T_n))$. Therefore, if a vertex v in V is of the form $v^0 + e_{I'}$ for some $v^0 \in V(T)$, v also belongs to $V(T)$ iff for each $i \in I(T)$, I' either contains $R(i)$ or is disjoint from it. The next lemma now follows just like Lemma B.3, its counterpart for $\mathcal{D}$.

Lemma B.5. A simplex of $\mathcal{D}$ is a principal simplex of $\mathcal{D}(T)$ iff there is an ordering $\pi^T : \{1, \dots, d_T\} \to I(T) \backslash I^*(T)$ such that the vertex set of D is of the form $v^0 \leqslant v^1 \leqslant \cdots \leqslant v^{d_T}$ where $v^0 \in V(T)$ and $v^l = v^{l-1} + e_{R(\pi^T(l))}$ for all $l > 0$.

For a vertex $v = x(\lambda)$ for some $\lambda \in \Lambda(T)$, if $v + e_{R(i^k(T_n))}$ belongs to $V(T)$, then $i^k(T_n) \notin T^*$ and $x^{-1}(v + e_{R(i^k(T_n))})$ is obtained from λ by adding the vector $\xi_{i^k(T_n)}^T \equiv p^{-1}(e_{i^k(T_n)} - e_{i^{k+1}(T_n)})$. The following theorem follows readily from the previous lemma.

Theorem B.6. A simplex L of $\mathcal{L}$ is a principal simplex of $\mathcal{L}(T)$ iff there is an ordering $\pi^T : \{1, \dots, d_T\} \to I(T) \backslash I^*(T)$ such that the vertex set of D is of the form $\{\lambda^0, \dots, \lambda^{d_T}\}$ where $\lambda^0 \in \Lambda(T)$ and for each $l > 0$, $\lambda^l = \lambda^{l-1} + \xi_{\pi^T(l)}^T$.

Like with Σ, a principal simplex in $\mathcal{L}(T)$ can be expressed as a pair (λ^0, π^T), where $\lambda^0 \in \Lambda(T)$ and π^T is an ordering of $T \backslash T^*$. For each T, let $\mathcal{L}^*(T)$ be the collection of the principal simplices of $\mathcal{L}(T)$ and let $\mathcal{L}^* = \cup_T \mathcal{L}^*(T)$.

The above constructions when applied to the space Σ_n (i.e. the case $N = 1$) yields a triangulation $\mathcal{L}_n$ of Σ_n with vertex set Λ_n and, for every subset T_n of S_n, a triangulation $\mathcal{L}_n(T_n)$ of the face $\Sigma_n(T_n)$. A principal simplex of $\mathcal{L}_n$ is given by a vertex λ_n^0 and an ordering $\pi_n : \{1, \ldots, |S_n| - 1\} \to S_n \backslash \{i_n^*\}$.

Observe that the principal simplices of $\mathcal{L}_n$ are the the projections of principal simplices of $\mathcal{L}$. Indeed, given a principal simplex (λ^0, π) of $\mathcal{L}$, let λ_n^0 be the projection of λ^0; and let π_n be the implied ordering for S_n, i.e. for any pair $i, j \in S_n \backslash \{i_n^*\}$ $\pi_n^{-1}(i) < \pi_n^{-1}(j)$ iff $\pi^{-1}(i) < \pi^{-1}(j)$. Then (λ_n^0, π_n) describes a principal simplex that is the projection of that given by λ^0 and π. Going the other way, given a collection of principal simplices (λ_n^0, π_n), one for each n, take λ^0 to be the product of the λ_n^0's, and choose any ordering π that respects the individual orderings in the sense that for each n and $i, j \in S_n \backslash \{i_n^*\}$, $\pi^{-1}(i) < \pi^{-1}(j)$ iff $\pi_n^{-1}(i) < \pi_n^{-1}(j)$. (λ^0, π) then describes a principal simplex in $\mathcal{L}$ whose projection to each Σ_n is the principal simplex we started with. Thus, the principal simplices of $\mathcal{L}_n$ are the projections of the principal simplices of $\mathcal{L}$. A similar statement holds for each face $\Sigma(T)$ of Σ in the sense that the principal simplices of $\mathcal{L}_n(T_n)$ is the projection of the principal simplices of $\mathcal{L}(T)$.

We conclude this section by introducing two important pieces of notation. For any simplex $L \in \mathcal{L}$, let $\Sigma_0(L)$ be the set of mixed strategies for player 0 whose support is contained in the vertex set of L. We denote by Σ_0^* the set of mixed strategies of player 0 whose support is a simplex in $\mathcal{L}$.

Let L be a principal simplex of $\mathcal{L}(T)$ for some $T \subsetneq S$ generated by a pair (λ^0, π). Suppose i is a strategy of some player n that is not contained in T_n. Let $T_n' = T_n \cup \{i\}$ and let T' be the product of T_n' with T_{-n}. There exists a unique principal simplex of $\mathcal{L}(T')$ that has L as a face. Denote this simplex by $L \vee i$. Likewise for a simplex L_n in $\mathcal{L}_n(T_n)$, $L_n \vee i$ denotes the the unique principal simplex of $\mathcal{L}_n(T_n')$ that has L_n as a face. (If L_n is the projection of L_n, $L_n \vee i$ is the projection of $L \vee i$.)

B.3. **The Coordinator's Best Replies.** The main task in this section is to derive results on when the vertices of a principal simplex of some face are all best replies. Since player 0's payoffs are given as a sum of his payoffs from bilateral interactions with the other players, we study this question for the case of a principal simplex of a player.

Fix some some player n. Recall that player 0's payoffs in his interactions with n are given by the function $\varphi_n(\lambda_n, \sigma_n) = p\lambda_n \cdot (\sigma_n - .5\lambda_n)$ for $\lambda_n \in \Lambda_n$ and $\sigma_n \in \Sigma_n$. A simple computation yields the following useful formula:

$$\varphi_n(\lambda_n, \sigma_n) - \varphi_n(\lambda_n + \delta_{i,j}, \sigma_n) = p^{-1} + \lambda_{n,i} - \lambda_{n,j} - \sigma_{n,i} + \sigma_{n,j} \tag{18}$$

for all $\lambda_n \in \Lambda_n$, $\sigma_n \in \Sigma_n$, and $i, j \in S_n$.

Lemma B.7. λ_n is a best reply to a strategy σ_n only if $|\sigma_{n,i} - \lambda_{n,i}| < p^{-1}$ for all $i \in S_n$. In particular, if λ_n is a best reply to σ_n, the support of λ_n is contained in that of σ_n.

Proof. The second statement follows trivially from the first, which we now prove. Suppose σ_n is such that either: (i) there exists i such that $\sigma_{n,i} - \lambda_{n,i} \geqslant p^{-1}$; or (ii) there exists j such that $\lambda_{n,j} - \sigma_{n,j} \geqslant p^{-1}$. Since obviously $\sigma_n \neq \lambda_n$, in case (i) there exists j such that $\sigma_{n,j} < \lambda_{n,j}$ and in case (ii) there exists i such at $\sigma_{n,i} > \lambda_{n,i}$. In both cases, therefore, $\lambda_n + \delta_{i,j}$ is a better reply than λ_n against σ_n, since

$$\varphi_n(\lambda_n, \sigma_n) - \varphi_n(\lambda_n + \delta_{i,j}, \sigma_n) = p^{-1} + \lambda_{n,i} - \lambda_{n,j} - \sigma_{n,i} + \sigma_{n,j} < 0\,. \tag{19}$$

□

Obviously, if λ_n is a best reply against σ_n it is at least as good a reply against σ_n as each each strategy of the form $\lambda_n + \delta_{i,j}$. The following lemma proves the converse.

Lemma B.8. Suppose λ_n is at least as good a reply against σ_n as $\lambda_n + \delta_{i,j}$ for each $i, j \in S_n$. (1) λ_n is a best reply against σ_n. (2) $\lambda'_n \neq \lambda_n$ is also a best reply against σ_n iff there exist subsets S_n^+ and S_n^- of S_n such that: (i) $\lambda'_n = \lambda_n + p^{-1}(e_{S_n^+} - e_{S_n^-})$; (ii) $\lambda_n + \delta_{ij}$ is a best reply against σ_n for all $i \in S_n^+, j \in S_n^-$.

Proof. We prove the lemma in two steps. First, we show that any $\lambda'_n \neq \lambda_n$ cannot be a best reply unless it is of the form given in (i) of statement 2 of the lemma. Second, we show that any $\lambda'_n \neq \lambda_n$ that is of the form given in 2(i) is not a strictly better reply than λ_n against σ_n and that is in fact as good a reply iff point (ii) of statement 2 holds.

Let $\lambda'_n \neq \lambda_n$ be an arbitrary strategy for player 0 that is not expressible in the form given by 2(i). Then either: (a) there exists i such that $\lambda'_{n,i} < \lambda_{n,i} - p^{-1}$; or (b) there exists j such that $\lambda_{n,j} > \lambda_{n,j} + p^{-1}$. In case (a) choose j such that $\lambda'_{n,j} \geqslant \lambda_{n,j} + p^{-1}$ and in case (b) choose i such that $\lambda'_{n,i} \leqslant \lambda_{n,i} - p^{-1}$. (Such a choice is possible since λ_n are λ'_n are different points in Λ_n.) In both cases, $\lambda'_{n,i} - \lambda'_{n,j} < \lambda_{n,i} - \lambda_{n,j} - 2p^{-1}$. Therefore, $\lambda'_n + \delta_{ij}$ is a strictly better reply against σ_n than λ'_n since

$$\varphi_n(\lambda'_n, \sigma_n) - \varphi_n(\lambda'_n + \delta_{i,j}, \sigma_n) = p^{-1} + \lambda'_{n,i} - \lambda'_{n,j} - \sigma_{n,i} + \sigma_{n,j} < -p^{-1} + \lambda_{n,i} - \lambda_{n,j} - \sigma_{n,i} + \sigma_{n,j} \leqslant 0\,, \tag{20}$$

where the last inequality follows from the fact that λ_n is a better reply than $\lambda_n + \delta_{j,i}$ against σ_n. Thus if λ_n is to be a best reply against σ_n it must be of the form given in 2(i).

Given now a strategy $\lambda'_n = \lambda_n + p^{-1}(e_{S_n^+} - e_{S_n^-})$, we can obviously take S_n^+ and S_n^- to be disjoint sets that, therefore, have the same cardinality. A direct computation gives us, then, that for any bijection $i \to j(i)$ between S_n^+ and S_n^-,

$$\varphi_n(\lambda_n, \sigma_n) - \varphi_n(\lambda'_n \sigma_n) = \sum_{i \in S_n^+} \left(\varphi_n(\lambda_n, \sigma_n) - \varphi_n(\lambda_n + \delta_{i,j(i)}, \sigma_n)\right). \tag{21}$$

Since λ_n is a better reply than each $\lambda_n + \delta_{ij}$, λ_n is a better reply than λ'_n. Finally, λ'_n is a best reply iff for some bijection $i \to j(i)$ (and hence for all bijections) $\lambda_n + \delta_{i,j(i)}$ is best reply for all i. □

Let L_n be a principal simplex of $\mathcal{L}_n(T_n)$ for some subset T_n of S_n. Let (λ_n^0, π) be the vertex-ordering pair generating L_n. And denote the vertex set by $\{\lambda_n^0, \ldots, \lambda_n^{d^{T_n}}\}$. Let $T_n^+(L_n)$ be the subset of coordinates i in $T_n \backslash \{ i_n^*(T_n) \}$ such that $\pi^{-1}(i) < \pi^{-1}(i-1)$, i.e., these are the coordinates i for which p^{-1} is first added to $\lambda_{n,i}^0$ before it is subtracted at a later stage. Let $T_n^-(L_n)$ be the complement of these coordinates in T_n: these coordinates first have a p^{-1} subtracted before it is added, if at all. Define $\sigma_n(L_n)$ by $\sigma_{n,i}(L_n) = \lambda_n^0 + p^{-1}(\alpha e_{T_n^+} - (1 - \alpha) e_{T_n^-})$, where $\alpha = |T_n^-|/(d^{T_n} + 1)$. We are now ready to state and prove the main result of this section.

Theorem B.9. For each principal simplex L_n of $\mathcal{L}_n(T)$:

(1) $\sigma_n(L_n)$ is the unique point in $\Sigma_n(T_n)$ against which all the vertices in L_n are best replies.
(2) For each $i \notin T_n$, the set of points in $\Sigma_n(T_n \cup \{ i \})$ against which the vertices of L_n are best replies is the interval $[\sigma_n(L_n), \sigma_n(L_n \vee i)]$.

Proof. Using the fact that $\varphi(\lambda^0, \sigma_n) - \varphi(\lambda^0 + \delta_{i,j}, \sigma_n) = p^{-1} - \sigma_{n,i} + \sigma_{n,j} + \lambda_{n,i}^0 - \lambda_{n,j}^0$, we see that $\lambda^0 + \delta_{i,j}$ and λ^0 are equally good replies against $\sigma_n(L_n)$ if $i \in T_n^+(L_n)$ and $j \in T_n^-(L_n)$; otherwise, λ^0 is a strictly better reply than $\lambda^0 + \delta_{i,j}$. Therefore, using the first statement of Lemma B.8, λ^0 is a best reply. Also, each vertex of L is of the form $\lambda^0 + p^{-1}(e_{\tilde{T}_n^+} - e_{\tilde{T}_n^-})$ for some $\tilde{T}_n^+ \subseteq T_n^+$ and $\tilde{T}_n^- \subseteq T_n^-$. Hence, using the second statement of Lemma B.8, all the vertices of L_n are best replies to $\sigma_n(L_n)$. To complete the proof of point (1) we will now show that there exists at most one point in $\Sigma_n(T_n)$ against which the vertices of L_n are equally good replies.

The set of $\sigma_n \in \Sigma_n$ against which the vertices of L_n are equally good replies satisfy the following system of linear equations:

$$\varphi_n(\lambda_n^l, \sigma_n) - \varphi_n(\lambda_n^{l+1}, \sigma_n) = 0 \quad 0 \leqslant l \leqslant d^{T_n} - 1 \tag{22}$$

$$\sum \sigma_{n,i} = 1. \tag{23}$$

It follows from simple linear algebra that there is at most one solution to this system in $\Sigma_n(T_n)$—viewing the variable σ_n as having $d^{T_n}+1$ coordinates, by dropping the coordinates not in T_n, the above system has $d^{T_n}+1$ linearly independent equations in $d^{T_n}+1$ variables. Hence, there exists at most one point in $\Sigma_n(L_n)$ against which the vertices of L_n are best replies.

We turn now to point (2). Fix $i \notin T_n$. Let C be the set of points in $\Sigma_n(T_n \cup \{ i \})$ against which the vertices of L_n are best replies. As before, elementary linear algebra involving the above system shows that C is a convex set whose dimension is at most 1. We know that C includes $\sigma_n(L_n)$; and, applying point (1) of this theorem to the vertices of $L_n \vee i$, we have that it also contains $\sigma_n(L_n \vee i)$. Therefore it contains the interval between the two points and in fact lies in the affine space A generated by this interval. Take a point σ_n in A that is outside this interval. We will show that it does not belong to C, which proves point (2). Since σ_n belongs to A, it is of the form $\beta\sigma_n(L_n)+(1-\beta)\sigma_n(L_n \vee i)$. Furthermore, because $\sigma_n(L_n)$ belongs to the boundary of $\Sigma_n(T_n \cup \{ i \})$, $\beta < 0$. Let λ_n be the vertex of $L_n \vee i$ that does not belong to L_n. Since i belongs to the support of λ_n but not to $\sigma_n(L_n)$, by Lemma B.7, λ_n is a worse best reply against $\sigma_n(L_n)$ than the vertices of L_n. Because it is a best reply against $\sigma_n(L_n \vee i)$, it is, therefore, a strictly better reply against σ_n than each vertex of L_n. Thus $\sigma_n \notin C$. □

Remark B.10. It need not even be true that $\sigma_n(L_n)$ belongs to L_n at all. For instance, suppose $|S_n| = p = 5$. Consider the simplex generated by the vertex $(1/5, 1/5, 1/5, 1/5, 1/5)$ and the ordering 2,3,4,1. $\sigma_n(L_n) = (8/25, 8/25, 3/25, 3/25, 3/25)$, which is an affine combination of the five vertices that uses weights $(-.2, .4, .4, -.2, .6)$, i.e. $\sigma_n(L_n)$ lies outside the convex hull of these five points. As is obvious from our formula, it is true, however, that the distance between $\sigma_n(L_n)$ and L_n, goes to zero as p goes to infinity.

B.4. **Selecting a Generic Set in $\mathcal{G}$.** For a game $G \in \mathcal{G}$ and a vector $g \in \mathbb{R}^I$, let $\tilde{G} \oplus g$ be the game where player 0's payoff function is still φ but where for each n, his payoff when he plays i and player 0 plays λ is $\tilde{G}_n(i,\lambda) + g_{n,i}$—thus, g is a vector of "bonuses" for the original players. Throughout this section and the next, g is a fixed vector in $\mathbb{R}^I$.

Our objective here is to choose a generic subset of $\mathcal{G}$ such that for each game G in this set, the best replies of all the original players are "non-degenerate" in the game $\tilde{G} \oplus \alpha g$ for all $\alpha \in \mathbb{R}$. This result helps us to carry out the procedure in the next section to compute an equilibrium of G.

Fix a principal simplex L with vertex set $\{ \lambda^0, \dots, \lambda^{d^T} \}$ in $\mathcal{L}(T)$ for some T. Recall that $\Sigma_0(L)$ is the set of mixed strategies whose support is contained in the vertex set of L. For $\sigma_0 \in \Sigma_0(L)$, $\sigma_{0,k}$ is the probability of vertex λ^k under σ_0.

Define $\theta(G, L)$ to be the set of $(\sigma_0, \beta) \in \Sigma_0(L) \times \mathbb{R}$ such that for each n, the strategies in T_n are all best replies to σ_0 in the game $\tilde{G} \oplus \beta g$. We now have the following important properties of $\theta(G, L)$ for a generic payoff function G.

Theorem B.11. There exists a closed, lower-dimensional semi-algebraic subset $\mathcal{G}_L$ of $\mathcal{G}$ such that for each $G \notin \mathcal{G}_L$, the set $\theta(G, L)$ is a closed interval with the following properties:

(1) Suppose (τ_0, β) belongs to the interior of $\theta(G, L)$. Then τ_0 belongs to the relative interior of $\Sigma_0(L)$; and for every player n, only the strategies in T_n are best replies against τ_0 in $\tilde{G} \oplus \beta g$.

(2) Suppose (τ_0, β) is an end point of $\theta(G, L)$ such that τ_0 is in the relative interior of $\Sigma_0(L)$. Then there is exactly one $i \in I \backslash I(T)$ that is also a best reply to τ_0 in the game $\tilde{G} \oplus \beta g$.

(3) Suppose (τ_0, β) is an end point of $\theta(G, L)$ such that τ_0 belongs to the relative boundary of $\Sigma_0(L)$. Then τ_0 belongs to the relative interior of one of the maximal proper faces of $\Sigma_0(L)$; and for each n, only the strategies in T_n are best replies against τ_0 in the game $\tilde{G} \oplus \beta g$.

Proof. For the purpose of this proof, we will view strategies in $\Sigma_0(L)$ as points in $\mathbb{R}^{d^T+1}$, rather than being in $\mathbb{R}^{\Lambda}$, by ignoring the coordinates in Λ that are not vertices of L. Consider now the following system of equations and inequalities in the variables $(G, \sigma_0, \beta, v) \in \mathcal{G} \times \mathbb{R}^{d^T+1} \times \mathbb{R} \times \mathbb{R}^{\mathcal{N}}$:

$$\psi_{n,i}(G, \sigma_0, \beta, v) \equiv G_n(i, \sigma_0) + \beta g_{n,i} - v_n = 0 \quad \forall n, i \in T_n \tag{24}$$

$$\psi_{n,i}(G, \sigma_0, \beta, v) \equiv G_n(i, \sigma_0) + \beta g_{n,i} - v_n \leqslant 0 \quad \forall n, i \notin T_n \tag{25}$$

$$\psi_{0,*}(G, \sigma_0, \beta, v) \equiv \sum_{k=0}^{d_T} \sigma_{0,k} - 1 = 0 \tag{26}$$

$$\psi_{0,k}(G, \sigma_0, \beta, v) \equiv \sigma_{0,k} \geqslant 0 \quad \forall 0 \leqslant k \leqslant d^T . \tag{27}$$

It has $d^T + N + 1$ equations and $I - N + 1$ inequalities in $N|S| + d^T + 2 + N$ variables. (Recall that $\sum_n |T_n| - N = d^T$ is the dimension of both $\Sigma(T)$ and L.) Let $\Theta(L)$ be the set of solutions to this system. For any game G, (σ_0, β) belongs to $\theta(G, L)$ iff there exists $v \in \mathbb{R}^{\mathcal{N}}$ such that (G, σ_0, β, v) belongs to $\Theta(L)$. Moreover, if we fix a game G, the above system is linear in the variables $(\sigma_0, \beta, v) \in \mathbb{R}^{d^T+1} \times \mathbb{R} \times \mathbb{R}^{\mathcal{N}}$. Therefore, the results stated in the theorem follow if we show that there exists a closed lower-dimensional semialgebraic subset $\mathcal{G}_L$ of $\mathcal{G}$ such that for $G \notin \mathcal{G}_L$, the following properties hold: (a) the set of $(G, \sigma_0, \beta, v) \in \Theta(L)$ that satisfies all inequalities strictly is 1-dimensional; (b) the set of $(G, \sigma_0, \beta, v) \in \Theta(L)$ that satisfy exactly one of the inequalities weakly is finite; (c) the set of $(G, \sigma_0, \beta, v) \in \Theta(L)$ that satisfy two or more inequalities weakly is empty.

For each nonnegative integer l, let $\Theta^l(L)$ be the subset of $\Theta(L)$ consisting of points (G, σ_0, β, v) for which exactly l of the inequalities are satisfied with equality. To obtain properties (a)-(c) above for generic G, our first task is to prove that $\Theta^l(L)$ is a manifold of dimension $N|S| + 1 - l$. To this end, fix $(G, \sigma_0, \beta, v) \in \Theta^l(L)$. Let J be the set of indices j such that $\psi_j(G, \sigma, \beta, v) = 0$ and choose a neighborhood U of (G, σ_0, β, v) such that ψ_j is strictly negative on U for all $j \notin J$. Define $\psi^J : \mathcal{G} \times \mathbb{R}^L \times \mathbb{R} \times \mathbb{R}^{\mathcal{N}} \to \mathbb{R}^J$ to be the function given by the collection $(\psi_j)_{j \in J}$ of coordinate functions. The domain of ψ^J has dimension $N|S| + d^T + N + 2$; while $|J| = \sum_n |T_n| + 1 + l = d^T + N + 1 + l$. Therefore, if we can show that the Jacobian of ψ^J has full rank then the implicit function theorem implies that the (G, σ_0, β, v) has a neighborhood V, which can be taken to be a subset of U, such that $(\psi^J)^{-1}(0) \cap V$ is homeomorphic to an open subset of $\mathbb{R}^{N|S|+1-l}$, which proves that $\Theta^l(L)$ is then a manifold of dimension $N|S| + 1 - l$. To prove that the Jacobian of ψ^J has full rank, observe that for each $(n, i) \in J$, and coordinate ψ_j of ψ^J, $\partial\psi_j / \partial G_n(i, s_{-n}) = 0$ for all $s_{-n} \in S_{-n}$ if $j \neq (n, i)$, while $\sum_{s_{-n}} \psi_i / \partial G_n(i, s_{-n}) = 1$. Thus, the derivative of $\psi_{n,i}$ is independent of the derivatives of ψ_j for all $j \neq (n, i)$. Since $\sum_k \sigma_{0,k} = 1$, there exists some k such that $(0, k) \notin J$. Therefore, the derivatives of the coordinate functions that correspond to restrictions on player 0's strategies are independent as well. Thus the derivative of ψ^J has full rank and $\Theta^l(L)$ is a manifold of dimension $N|S| + 1 - l$.

Let $\mathrm{proj} : \Theta(L) \to \mathcal{G}$ be the natural projection. For each $l > 1$, let $\mathcal{G}_L^l$ be the closure of $\mathrm{proj}(\Theta^l(L))$. Then, $\mathcal{G}_L^1$ is a semialgebraic set of dimension at most $N|S| + 1 - l < N|S|$. For each $G \notin \mathcal{G}_L^l$ for $l > 1$, property (c) above holds.

For $l = 0, 1$, applying the generic local triviality theorem to the projections from $\Theta^l(L) \to \mathcal{G}$ we get that there exists closed semialgebraic subset $\mathcal{G}_L^l$ of $\mathcal{G}$ such that for each component C of $\mathcal{G} \backslash \mathcal{G}_L^l$, there exists a semialgebraic set F_C such that $\mathrm{proj}^{-1}(C) \cap \Theta^l(L)$ is homeomorphic to $C \times F_C$ under a map that sends $\{ G \} \times F_C$ to $\mathrm{proj}^{-1}(G) \cap \Theta^l(L)$. Since C is open in $\mathcal{G}$, $\mathrm{proj}^{-1}(C) \cap \Theta^l(L)$ is an open subset of the semialgebraic manifold $\Theta^l(L)$ and hence has dimension $N|S| + 1 - l$. Therefore, F_C and hence also $\mathrm{proj}^{-1}(G) \cap \Theta^l(L)$ has dimension $1 - l$. If $G \notin \mathcal{G}_L^0$, then property (a) holds, while if $G \notin \mathcal{G}_L^1$, property (b) holds. Let $\mathcal{G}_L$ be the union of the sets $\mathcal{G}_L^l$ for $l \geqslant 0$. Then properties (a)-(c) hold for any $G \notin \mathcal{G}_L$. $\square$

For a game $G \in \mathcal{G}$, let $\theta(G) = \cup_{L\in\mathcal{L}^*}\theta(G, L)$. Let $\mathcal{G}^* = \mathcal{G}\setminus \cup_{L\in\mathcal{L}^*} \mathcal{G}_L$. We now have the following theorem concerning the structure of $\theta(G)$, which implies that it is a 1-dimensional piecewise-linear manifold.

Theorem B.12. Let G be a game in $\mathcal{G}^*$. Let $(\sigma, \sigma_0, \alpha) \in \theta(G, L)$ where $L \in \mathcal{L}^*(T)$ for some T. Let $L' \neq L$ be another principal simplex in $\mathcal{L}^*$.

(1) If (σ_0, α) belongs to the interior of $\theta(G, L)$ then it does not belong to $\theta(G, L')$.
(2) If (σ_0, α) belongs to the boundary of $\theta(G, L)$ and σ_0 belongs to the interior of $\Sigma_0(L)$, then it belongs to $\theta(G, L')$ iff $L' = L \vee i$, where i is as in Statement 2 of Theorem B.11.
(3) If (σ_0, α) belongs to the boundary of $\theta(G, L)$ and σ_0 belongs to the interior of $\Sigma_0(K)$ where K is a maximal proper face of L, then (σ_0, α) belongs to $\theta(G, L')$ iff either $L = K \vee i$ for some $i \notin I(K)$ or L' belongs to $\mathcal{L}^*(T)$ and has K as a maximal proper face.

Proof. The sufficiency proof of Statements 2 and 3 are obvious. We prove the rest of the theorem now. The basic observation that drives these other conclusions of the theorem is the following. Suppose (σ_0, α) belong to $\theta(G, L')$, then $I(T')$ is contained in the set of the best replies (of all the players taken together) against σ_0 in the game $\tilde{G} \oplus \alpha g$.

Suppose (σ_0, α) belongs to $\theta(G, L) \cap \theta(G, L')$; and suppose it satisfies the condition of either Statement 1 or 3, i.e. σ_0 either belongs to the interior of $\Sigma_0(L)$ or to the interior of a maximal proper face $\Sigma_0(K)$ of $\Sigma_0(L)$. Then by the previous theorem, only the strategies in $I(T)$ are optimal against σ_0 in the game $\tilde{G} \oplus \alpha g$. The above observation now tells us that $I(T') \subseteq I(T)$, i.e. $T' \subseteq T$. Therefore L' is a simplex in $\mathcal{L}(T)$. Since L is a principal simplex of $\mathcal{L}(T)$, $\Sigma_0(L') \cap \Sigma_0(L)$ is therefore a proper face of $\Sigma_0(L)$. In particular σ_0 cannot belong to the interior of $\Sigma_0(L)$ and has to belong to the interior of a maximal proper face $\Sigma_0(K)$ of $\Sigma_0(L)$, thus proving Statement 1. Statement 3 follows as well, from the fact that $\Sigma_0(L) \cap \Sigma_0(L')$, which contains σ_0 and is a face of $\Sigma_0(L')$, must now equal $\Sigma_0(K)$.

Suppose finally that (σ_0, α) belongs to $\theta(G, L) \cap \theta(G, L')$ and satisfies the condition of Statement 2. Since σ_0 belongs to the interior of $\Sigma_0(L)$, we have that $\Sigma_0(L)$ is a face of $\Sigma_0(L')$; and our observation above shows that $I(T') \subseteq I(T) \cup \{ i \}$. Combining these two facts, we have $I(T') = I(T) \cup \{ i \}$, i.e. $L' = L \vee i$. □

B.5. **Paths of the Algorithm.** Take a game G that belongs to $\mathcal{G}^*$. Let E^* be the set of $(\sigma, \sigma_0, \beta) \in \Sigma \times \Sigma_0^* \times \mathbb{R}$ such that (σ, σ_0) is an equilibrium of $\tilde{G} \oplus \beta g$. We show that E^* is essentially a piecewise linear 1-dimensional manifold, which provides the basis for the decomposition algorithm to compute equilibria of G successfully.

Lemma B.13. $(\sigma, \sigma_0, \alpha)$ belongs to E^* only if (σ_0, α) belongs to $\theta(G)$.

Proof. Let $(\sigma, \sigma_0, \alpha)$ be a point in E^*. Let L^* be the simplex spanned by the support of σ_0. There exists a unique face $\Sigma(T)$ of Σ whose relative interior contains the relative interior of the simplex L^*. Let L be a principal simplex of $\mathcal{L}(T)$ that has L^* as a face. We will now show that (σ_0, α) belongs to $\theta(G, L)$.

Since the relative interior of L^* is contained in that of $\Sigma(T)$, for each player n and each $t_n \in T_n$, there exists a vertex of L^* whose projection to Σ_n has t_n in its support. Since the support of σ_0 is the vertex set of L^*, and since it is a best reply to σ for player 0, it now follows from Lemma B.7 that T_n is included in the support of σ_n for each n. Therefore, each strategy in T_n is a best reply for player n against σ_0 in the game $\tilde{G} \oplus \alpha g$. It now follows from its definition that $\theta(G, L)$ contains (σ_0, α). □

Thanks to the above lemma, and Theorem B.11, we need to focus only on strategies for player 0 whose support is either a principal simplex or a maximal proper face of a principal simplex in some face of Σ. If L is a principal simplex in a face of Σ, denote by $\sigma(L)$ the mixed strategy profile in Σ where each player n plays the strategy $\sigma_n(L_n)$. The following lemma helps provide a near complete description of E^*.

Lemma B.14. Fix $(\sigma, \sigma_0, \alpha) \in E^*$ and let L be a principal simplex of a face $\Sigma(T)$ of Σ such that $(\sigma_0, \alpha) \in \theta(G, L)$.

(1) If (σ_0, α) belongs to the interior of $\theta(G, L)$ then $\sigma = \sigma(L)$.
(2) If (σ_0, α) belongs to the boundary of $\theta(G, L)$ but σ_0 belongs to the interior of $\Sigma_0(L)$ then $\sigma \in [\sigma(L), \sigma(L \vee i)]$, where $i \in I \backslash I(T)$ is as in Statement 2 of Theorem B.11.
(3) If (σ_0, α) belongs to the boundary of $\theta(G, L)$ and σ_0 belongs to the interior of a maximal proper face $\Sigma_0(K)$ of $\Sigma_0(L)$ where $L = K \vee i$ for some $i \notin I(K)$, then $\sigma \in [\sigma(K), \sigma(L)]$.

Proof. If (σ_0, α) belongs to the interior of $\theta(G, L)$ then the support of σ_n is contained in T_n for each n by Statement 1 of Theorem B.11. Since σ_0 is an interior point of $\Sigma_0(L)$, by Statement 1 of Theorem B.9 we have $\sigma_n = \sigma_n(L_n)$ for each n, which proves the first statement.

Suppose (σ_0, α) satisfies the conditions of Statement 2. Let n the player whose strategy set includes i. By Statement 2 of Theorem B.11, the support of σ_m for each $m \neq n$ is contained in T_m while for player n it is contained in $T_n \cup \{ i \}$. Since σ_0 belongs to the interior of $\Sigma_0(L)$, therefore, by Theorem B.9, $\sigma_n \in [\sigma_n(L_n), \sigma_n(L_n \vee i)]$ and $\sigma_m = \sigma_m(L_m)$ for $m \neq n$, which proves Statement 2.

Finally, Statement 3 follows by applying Statement 2 to the simplex K. □

Combining the results of the preceding two lemmas, we have characterized the exact nature of equilibria in E^* except for one case: equilibria of the form $(\sigma, \sigma_0, \alpha)$ where (σ_0, α) belongs to $\theta(G, L)$ and $\theta(G, L')$ for two simplices $L \neq L'$ whose intersection is a maximal proper face of each. In this case, $[\sigma(L), \sigma(L')] \times \{ (\sigma_0, \alpha) \}$ is contained in E^* but there could be equilibria where σ does not belong to this interval. If we exclude these "extraneous solutions" we get a manifold, as we show now.

Let E be the set of $(\sigma, \sigma_0, \alpha) \in E^*$ such that if σ_0 belongs to a maximal proper face of two different simplices L and L' in $\mathcal{L}^*$, then $\sigma \in [\sigma(L), \sigma(L')]$.

Theorem B.15. E is a 1-dimensional piecewise-linear manifold without boundary.

Proof. Let $(\sigma, \sigma_0, \alpha)$ be a point in E. Then, there exists T and $L \in \mathcal{L}^*(T)$ such that $(\sigma_0, \alpha) \in \theta(G, L)$.

If (σ_0, α) belongs to the interior of $\theta(G, L)$, by Statement 1 of Theorem B.12, $\theta(G, L)$ is a neighborhood of (σ_0, α) in $\theta(G)$. By Statement 1 of Lemma B.14, then, $\{ \sigma(L) \} \times \theta(G, L)$ is a neighborhood of $(\sigma, \sigma_0, \alpha)$ in E.

If (σ_0, α) belongs to the boundary of $\theta(G, L)$ but σ_0 belongs to the interior of $\Sigma_0(L)$, by Statement 2 of Theorem B.12, $\theta(G, L) \cup \theta(G, L \vee i)$ is a neighborhood of (σ_0, α) in $\theta(G)$. Therefore, from Statements 1 and 2 of Lemma B.14 we get that $(\{ \sigma(L) \} \times \theta(G, L)) \cup ([\sigma(L), \sigma(L \vee i)] \times \{ (\sigma_0, \alpha) \}) \cup (\{ \sigma(L \vee i) \} \times \theta(G, L \vee i))$ is a neighborhood of $(\sigma, \sigma_0, \alpha)$ in E.

If (σ_0, α) belongs to the boundary of $\theta(G, L)$ and σ_0 belongs to the interior of a face $\Sigma_0(K)$ where K is a maximal face of $\Sigma_0(L)$, then by Statement 3 of Theorem B.12, $\theta(G, L) \cup \theta(G, K)$ is a neighborhood of $(\sigma_0, \alpha) \in \theta(G)$. Also, either $K \in \mathcal{L}^*$ or $K = L \cap L'$ for some $L' \in \mathcal{L}^*(T)$. Using Statement 3 of Lemma B.14 in the former case and our assumption on E in the latter, we get that $(\{ \sigma(L) \} \times \theta(G, L)) \cup ([\sigma(L), \sigma(K)] \times \{ (\sigma_0, \alpha) \}) \cup (\{ \sigma(K) \} \times \theta(G, K))$ is a neighborhood of (σ_0, α) in E. □

Suppose for each n there exists $i_n \in S_n$ such that $g_{n,i_n} > g_{n,i}$ for all $i \neq i_n$ in S_n. Then for all large α the game $\tilde{G} \oplus \alpha g$ has a unique equilibrium in which each player n plays i_n and the coordinator proposes this pure strategy profile. Thus there is a unique component of E that contains this equilibrium point for large α. Such an equilibrium can be used to initialize the algorithm, which then guarantees convergence.

Appendix C. Results of Test Problems

For each triple (N, m, K), Table 3 reports for 100 random examples with payoffs uniformly distributed on $[-1, 1]$ the average total CPU time (in megacycles) required to reach an equilibrium, and similarly, Table 4 reports the averages of (1) the portion of the total time devoted to computing payoffs (in megacycles), (2) the number of times that payoffs were computed (in thousands), and (3) the number of pivot operations (in thousands). Table 5 reports the worst case (in terms of total megacycles) encountered in each set of 100 examples. For $K = 40$, Tables 6 and 7 show for selected cases the histograms of total CPU time among the 100 examples, first on a relative scale and then on an absolute scale — as mentioned in the text, the tails of these distributions are quite long.

Table 3. Average Total CPU Megacycles for 100 Random Examples

N	K	m: 2	3	4	5	6	7	8	9	10	11	12
2	10	0.0	0.1	0.1	0.1	0.1	0.2	0.2	0.2	0.2	0.5	4.7
	20	0.1	0.1	0.9	0.2	0.2	0.3	0.4	0.4	1.4	6.0	1.0
	40	0.1	0.1	0.2	0.3	0.4	0.6	0.8	0.8	6.3	1.7	7.3
3		0.1	0.2	0.3	0.5	0.7	1.5	2.1	3.7	3.6	7.8	11.4
		0.3	0.3	0.5	0.9	1.3	3.1	5.1	5.6	8.1	13.8	19.1
		0.2	0.5	0.9	1.8	2.6	6.2	7.8	15.3	12.2	26.2	39.1
4		0.1	0.4	1.0	1.7	4.5	7.9	12.4	24.3	39.5	89.1	101.6
		0.2	0.7	1.9	2.9	10.6	13.5	21.9	43.5	58.5	94.4	176.8
		0.4	1.4	3.8	5.8	18.1	25.6	53.8	75.9	130.8	182.6	374.2
5		0.2	0.8	2.8	8.6	15.2	53.8	73.4	196.9	334.1		
		0.4	1.6	5.4	16.9	31.6	101.2	120.0	209.2	718.1		
		0.9	3.1	10.7	28.3	50.9	160.6	243.7	475.3			
6		0.5	2.6	9.7	26.1	74.8	218.9					
		0.9	5.0	15.3	42.4	117.3	271.8					
		1.7	9.8	39.2	90.9	206.2	458.2					
7		0.8	5.6	29.9	86.8							
		1.5	10.8	48.9	136.6							
		3.0	21.3	91.0	308.2							
8		1.4	10.4	54.7								
		2.7	19.1	121.2								
		5.3	43.4	210.9								
9		2.1	25.0	192.2								
		4.1	46.1	324.2								
		8.0	88.8	739.7								
10		4.1	61.3									
		7.8	125.7									
		15.6	204.8									
11		4.9	100.4									
		10.9	257.0									
		21.6	691.6									
12		9.8										
		19.1										
		36.8										

Table 4. Payoff and Pivot Calculations

For $K = 40$, each cell shows for 100 random examples the averages of:
(1) Total time to compute payoffs (in megacycles),
(2) Number of times payoffs computed (in thousands),
(3) Number of pivot operations (in thousands).

N	m:	2	3	4	5	6	7	8	9	10	11	12
2	(1)	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.4	0.1	0.4
	(2)	0.1	0.1	0.1	0.2	0.3	0.4	0.5	0.5	4.0	1.0	3.5
	(3)	0.1	0.2	0.3	0.4	0.5	0.6	0.8	0.8	5.9	1.5	6.6
3		0.0	0.1	0.1	0.2	0.3	0.6	0.7	1.4	1.1	2.2	3.3
		0.1	0.3	0.6	1.0	1.3	2.9	3.5	6.9	5.0	9.8	13.4
		0.2	0.5	0.8	1.4	2.0	4.2	5.0	10.3	6.9	13.4	18.1
4		0.1	0.2	0.5	0.7	2.2	3.1	6.5	9.6	17.5	26.4	36.0
		0.2	0.7	1.8	2.5	6.8	9.0	18.0	22.0	34.4	42.2	55.1
		0.3	1.1	2.5	3.4	9.1	12.0	23.7	28.5	44.5	54.8	70.5
5		0.2	0.5	1.7	4.4	8.4	29.8	49.2	116.0			
		0.4	1.3	3.9	9.0	14.1	36.0	47.5	73.7			
		0.6	1.8	5.2	11.7	18.1	45.1	59.2	91.1			
6		0.3	1.8	7.1	18.4	53.0	139.4					
		0.6	3.2	10.9	21.3	34.4	55.8					
		0.9	4.2	15.8	27.8	42.3	67.9					
7		0.7	4.6	22.5	96.6							
		0.9	5.6	17.5	39.1							
		1.4	7.2	21.6	47.3							
8		1.3	10.2	67.1								
		1.4	9.2	26.5								
		1.9	11.6	32.1								
9		2.0	25.8	304.8								
		1.8	13.6	43.1								
		2.4	16.6	51.0								
10		4.3	68.0									
		2.9	23.3									
		3.8	29.9									
11		6.3	185.6									
		3.5	28.5									
		4.5	33.8									
12		11.3										
		5.0										
		6.3										

Table 5. Maximum Total CPU Megacycles in Each Set of 100 Examples

N	K	m: 2	3	4	5	6	7	8	9	10	11	12
2	10	0	0	0	1	1	1	1	2	1	4	422
	20	0	0	82	1	2	2	2	4	90	509	8
	40	0	1	1	2	3	3	5	9	528	17	544
3		0	1	2	3	5	10	14	129	27	80	172
		18	1	3	9	10	31	121	115	131	158	172
		1	3	6	19	20	61	43	631	76	308	463
4		0	2	6	14	30	68	127	309	763	1289	945
		1	4	11	24	136	135	151	631	748	971	1853
		2	8	21	49	156	189	950	587	1454	2215	12946
5		1	4	15	83	88	297	1017	2620	4513		
		2	7	33	238	350	552	1127	1563	3286		
		4	15	64	189	364	1321	2134	2741			
6		2	9	69	196	562	5296					
		5	18	76	282	752	1209					
		11	35	879	1073	1870	5201					
7		5	40	513	780							
		9	75	482	1197							
		17	161	650	2544							
8		6	62	233								
		11	175	1277								
		22	893	1148								
9		7	164	2781								
		30	178	4163								
		36	383	8119								
10		23	1445									
		40	1426									
		85	1440									
11		27	1816									
		66	3812									
		132	16270									
12		59										
		105										
		187										

Table 6. Examples of Histograms of Cycles (Relative Scale)

In each row the width of each cell is 5% of the maximum number of cycles. Each entry shows the number of examples requiring the corresponding number of megacycles.

N	m																				
4	12	89	6	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
5	9	17	8	6	2	2	4	1	0	1	0	0	1	0	0	1	0	1	1	0	1
6	7	54	15	12	6	6	3	1	1	0	1	0	0	0	0	0	0	0	0	0	1
7	5	42	17	13	10	6	7	0	0	1	1	0	0	0	0	0	0	2	0	0	1
8	4	26	20	16	9	9	3	3	3	1	1	1	2	0	0	0	1	2	1	1	1
9	4	52	21	12	5	2	2	3	1	0	0	0	0	0	0	0	0	0	0	0	2
10	3	42	18	13	5	3	4	5	2	2	0	2	1	1	0	0	0	0	0	1	1
11	3	87	6	3	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Table 7. Examples of Histograms of Cycles (Absolute Scale)

In each row the width of each cell is 50 megacycles.

N	m																					> 1000
4	12	29	20	15	7	1	3	2	4	3	4	0	1	0	1	0	0	2	0	0	0	8
5	9	13	4	0	4	2	5	0	3	0	0	2	1	0	2	3	0	1	0	0	0	6
6	7	21	14	10	5	3	4	2	3	3	1	5	3	4	1	1	1	4	0	1	1	13
7	5	20	16	12	6	5	6	3	7	3	2	5	3	1	5	1	0	0	0	0	0	5
8	4	19	24	15	13	2	8	3	2	3	1	1	0	2	1	0	0	0	1	2	1	2
9	4	14	4	6	10	6	4	6	2	2	6	3	2	3	0	2	2	4	2	0	1	21
10	3	26	24	11	10	4	3	2	4	1	4	2	3	0	0	0	2	1	1	0	0	2
11	3	10	12	7	11	8	3	8	5	3	2	3	5	2	2	2	4	2	0	0	0	11

Appendix D. The APL Program for the Decomposition Algorithm

Function DA (main function for computing equilibrium of G when sums = K)

```
sigma←K DA G
⍝∇ DA - Compute approximate equilibrium of G
⍝∇      using Decomposition Algorithm with simplicial subdivision
⍝∇ G is 1+N dimensional payoff array from profiles of N pure strategies
⍝∇       [or use a function G in Pay]
⍝∇ Each player n's simplex = nonnegative integer points summing to K[n]

⍝⋆⋆   Initialize: Set up parameters
    Abort←0 ⋄ ts←⎕ts ⋄ N←(⍴G)[1] ⋄ K←N⍴K ⋄ m←1↓⍴G ⋄ M←+/m ⋄ L←0,+\m ⋄ Le←0⍴0
    NoEq←0 ⋄ equi←0 6⍴0 ⋄ n←1
I1: Le←Le,⊂L[n]+⍳m[n] ⋄ →(N≥n←n+1)/I1 ⋄ BF←0 ⋄ →(M=⍴g)/L0 ⋄ g←0⍴0 ⋄ n←1
I2: g←g,m[n]↑1 ⋄ →(N≥n←n+1)/I2 ⋄ →L0 ⍝⍝ Default g
⍝    Set up initial tableaus
L0: D←1,N⍴1 ⋄ T←LS←P←V←s←ss←a←0⍴0 ⋄ Q←(N,M)⍴0 ⋄ NoPivots←0 ⋄ n←1
⍝    Find n's best-reply to g
L1: j←n⊃Le ⋄ Q[n;j]←1 ⋄ ss←ss,L[n]+i←(⍒g[j])[1] ⋄ pn←m[n]⍴0 ⋄ pn[i]←K[n]
    s←s,pn ⋄ P←P,⊂1⍴i ⋄ LS←LS,i ⋄ V←V,⊂pn ⋄ a←a,⊂(2×K[n]×pn)-+/pn⋆2 ⋄ →(N≥n←n+1)/L1
    An←V ⋄ Seq←2 0⍴0 ⋄ Inc←N⍴⊂0⍴0 ⋄ V←(1,N)⍴V
⍝    Fill tableaus for 0's initial proposal
    v←(M,1)⍴ Payoff s ⋄ T←T,⊂t←(((v⍪1),((-⍉Q)⍪0)),(g,0)),(M⍴0),1 ⋄ n←1
L2: T←T,⊂((((1,m[n])⍴n⊃a)⍪1),(¯1,0)),(0,K[n]) ⋄ →(N≥n←n+1)/L2
⍝    Specify labels and lex ordering
    gcol←0 ⋄ Last←1000+0,⍳N ⋄ V0←9600+0,⍳N ⋄ v0←9700+0,⍳N
    row←⊂r←(-⍳M),-V0[1] ⋄ col←⊂c←(⊂An),v0[1+⍳N],gcol,Last[1] ⋄ n←1
L3: row←row,⊂(An[n]),-v0[1+n] ⋄ col←col,⊂(n⊃Le),V0[1+n],Last[1+n]
    →(N≥n←n+1)/L3 ⋄ lex←⊂Last[1],gcol,(-⍳M),(⊂An) ⋄ n←1
L4: lex←lex,⊂Last[1+n],(⍳m[n]) ⋄ →(N≥n←n+1)/L4
    rowI←row ⋄ colI←col ⍝ initial configuration for use in Adj & ADJ
```

```
⍝    Do initial pivots
    1 PivotD (⊂An),-V0[1] ⋄ k←1+n←1
L5: 1 PivotD v0[k],-ss[n] ⋄ k PivotD ss[n],-v0[k] ⋄ k PivotD V0[k],An[n]
    k←1+n←n+1 ⋄ →(N≥n)/L5 ⍝⋄ display ⋄ 1÷0

⍝★★   Lemke-Howson algorithm
⍝     Identify first pivot in T[1] : uses gcol
     k←1 ⋄ r←k⊃row ⋄ pc←(k⊃T)[;(k⊃col)⍳Last[k],gcol] ⋄ p←((∧/pc<0)∧~r∊V0,v0)/⍳⍴r
     →(0=⍴p)/Fin ⋄ j←p[(⍋÷/pc[p;])[1]] ⋄ PC←gcol ⋄ PR←r[j] ⋄ →Pvt

⍝★    Pivot in T[k] : uses k,PC,PR
Pvt: r←k⊃row ⋄ c←k⊃col ⋄ pr←r⍳PR ⋄ pc←c⍳PC ⋄ t←k⊃T ⋄ prow←t[pr;] ⋄ pcol←t[;pc]
     sp←×pvt←prow[pc] ⋄ t←((t×pvt)-pcol∘.×prow)÷D[k]×sp ⋄ t[pr;]←sp×prow
     t[;pc]←-sp×pcol ⋄ t[pr;pc]←sp×D[k] ⋄ x←r[pr] ⋄ r[pr]←c[pc] ⋄ c[pc]←x
     row[k]←⊂r ⋄ col[k]←⊂c ⋄ T[k]←⊂t ⋄ D[k]←|pvt ⋄→(MaxPivots<NoPivots←NoPivots+1)/Fin
     →((D[k]>1E¯9)∧(D[k]<1E9))/P1 ⋄ T[k]←⊂t←t÷D[k] ⋄ D[k]←1 ⋄ →P1 ⍝Reduce D[k] if big

⍝     Purge old row or column : uses PR PC
P1:  →((1<k)∧(1<≡PC))/Pur ⋄ →P2
⍝     Purge the row of the now-basic BF-slack if that proposal to n was dropped
Pur: r←k⊃row ⋄ i←~r∊PC ⋄ T[k]←⊂i⌿k⊃T ⋄ row[k]←⊂i/r
     rowI[k]←⊂(k⊃rowI)~PC ⋄ →P3
P2:  →((1=k)∧(1<≡PR))/PUR ⋄ →P3
⍝     Purge the column of old PR from T[1] if it was a vertex now dropped : uses PR
PUR: c←1⊃col ⋄ j←~c∊PR ⋄ T[1]←⊂j/1⊃T ⋄ col[1]←⊂j/c
     colI[1]←⊂(1⊃colI)~PR ⋄ →P3
P3:  →(2>≡1⊃row)/0
Out: →(1<≡PR)/FPC ⋄ →((PR≢gcol)∧Abort≠1)/FPC ⋄ →Fin

⍝★    Find next pivot column PC : uses BF,oldPR
FPC:  r←⍴oV←Simplex ⋄ r←r[1] ⋄ (oA oS oL oI oP)←(An Seq LS Inc P)
      k←1 ⋄ →(BF=0,1)/FC0,ADJ ⋄ 1÷0

FC0: →(1<≡PR)/New ⋄ PC←-PR ⋄ n←¯1+k←((PC∊¨col)/⍳N+1)[1] ⋄ →(n=0)/FPR ⋄ →NewD
```

```
⍝     Find new vertex np of adjacent simplex if adding dimension (n,j),j=PC-L[n]
⍝     : uses n,PC
NewD:j←PC-L[n] ⋄ q←n⊃oI ⋄ p←n⊃oP ⋄ f←⌊/p ⋄ l←⌈/p
     →((j<f),(j>l))/LF,LL ⋄ →LM

     ⍝↓ If j<f then adjoin n,j as last increment
LF:  Seq←oS,(n,j) ⋄ jp←r+1 ⋄ →LE

     ⍝ If f<j<l then insert increment n,j into Seq
LM:  s←((oS[1;]∊n)∧(oS[2;]∊⌈/(q<j)/q))/⍳r-1 ⋄ s←s[1]
     Seq←((2,s-1)↑oS),(2 1⍴n,j),(0,s-1)↓oS ⋄ jp←s+1 ⋄ →LE

     ⍝↓ If j>l then adjoin (n,l) as first increment and revise An & LS
LL:  pn←n⊃An ⋄ pn[l,j]←pn[l,j]+¯1,1
     An[n]←⊂pn ⋄ Seq←(n,l),oS ⋄ LS[n]←j ⋄ jp←1 ⋄ →LE

LE:  V←Simplex ⋄ np←V[jp;] ⋄ →Adj

⍝     Find new NF vertex np [if any] when vertex V[ip;], p=⊃PR is dropped: uses oldPR
New: p←⊃PR ⋄ v←∧/¨oV=(⍴oV)⍴p ⋄ ip←((∧/v)/⍳r)[1] ⍝ ip = row of dropped vertex

⍝     Find new vertex np [if any] when simplex flipped by dropping p: uses oA,oS,oL,ip
NV:  n←oS[1;1⌈ip-1] ⋄ q←n⊃oI ⋄ →(ip=1,r)/N1,N3 ⋄ →N2

    ⍝↓ First increment rotated to last, or first find face str[l]=0 if reached
N1:  An←oV[2;] ⋄ n←oS[1;1] ⋄ q←n⊃oI
     →((oS[2;1]=q[¯1+q⍳oL[n]])∧((n⊃oA)[oL[n]]=1))/N1F ⋄ →N1V
N1F: Seq←0 1↓oS ⋄ i←LS[n] ⋄ LS[n]←oS[2;1] ⋄ →DF ⍝i=n's dropped dimension
    ⍝↑ Face with str[LS[n]]=0 when decrement is to LS[n], if zero after Anchor => np<0

N1V: Seq←1⌽oS ⋄ i←r ⋄ →N4 ⍝ i=n's dropped BF proposal

    ⍝↓ Succesive increments switched, or first find face str[j[1]]=0 if reached
```

```
N2:  j←⊃S[2;(ip-1),ip]
     →((n=⊃S[1;ip])∧(((q⍳j[1])=1+(q⍳j[2]))∧((n⊃⊃A)[j[1]]=0)))/N2F ⋄ →N2V
N2F: Seq←⊃S[;(⍳r-1)~ip-1] ⋄ i←j[1] ⋄ →DF
     ⍝↑ Face with str[j[1]]=0 when n=n',j=j'+1 and (n,j) is zero in Anchor => np<0
N2V: Seq[;(ip-1),ip]←Seq[;ip,ip-1] ⋄ i←ip ⋄ →N4

    ⍝↓ Last increment rotated to first, or first find face str[f]=0 if reached
N3:  j←⊃S[2;r-1] ⋄ →((j=q[1])∧((n⊃⊃A)[q[1]]=0))/N3F ⋄ →N3V
N3F: Seq←0 ¯1↓⊃S ⋄ i←j ⋄ →DF
     ⍝↑Face with str[j]=0 when j=n's first, which is zero in Anchor => np<0
N3V: pn←n⊃An←⊃A ⋄ jj←q[1+q⍳j] ⋄ pn[j,jj]←pn[j,jj]+¯1,1
     An[n]←⊂pn ⋄ Seq←¯1⌽⊃S ⋄ i←1 ⋄ →N4

N4:  V←Simplex ⋄ np←V[i;]
     →(∧/⊃∧/¨np≥¨0)/D0 ⋄ 1÷0 ⍝checked in Simplex too

DF:  V←Simplex ⋄ np←⍬ ⋄ k←1+n ⋄ BF←0 ⋄ PC←p[n] ⋄ PR←L[n]+i ⋄ →Pvt

⍝     Find BF proposal dropped [if any] by dropping NF proposal p←⊃PR : uses v,p
D0:  s←+⌿v ⋄ →(∧/s>1)/ADJ ⋄ k←1+n←(⍋s)[1] ⋄ PC←p[n] ⋄ →Adj

⍝     Adjoin new BF row to T[k] : uses n,np ⍝ Form of <a> assumes f(x)=+/x*2
Adj: pn←n⊃np ⋄ a←(2×K[n]×pn)-+/pn*2 ⋄ rnew←a,¯1 0
     t←k⊃T ⋄ r←k⊃row ⋄ c←k⊃col ⋄ rI←k⊃rowI ⋄ cI←k⊃colI
     x←(r∊cI)/r ⋄ new←rnew[cI⍳x]+.×-t[r⍳x;]
     x←(y←c∊cI)/c ⋄ new←new+y\rnew[cI⍳x]×D[k]
     T[k]←⊂new⍪t ⋄ row[k]←⊂(⊂pn),r ⋄ rowI[k]←⊂(⊂pn),rI ⋄ PR←⊂pn ⋄ BF←1 ⋄ →Pvt

⍝     Adjoin new NF payoff column to T[1] : uses np
ADJ: →(1<≡np)/A1 ⋄ 1÷0 ⍝New skipped if BF=1 since np inherited from previous iteration
A1:  v←Payoff s←,⊃np ⋄ cnew←v,1
     t←1⊃T ⋄ r←1⊃row ⋄ c←1⊃col ⋄ rI←1⊃rowI ⋄ cI←1⊃colI
     x←(c∊rI)/c ⋄ new←t[;c⍳x]+.×cnew[rI⍳x]
     x←(y←r∊rI)/r ⋄ new←new+y\cnew[rI⍳x]×D[1]
```

```
     T[1]←⊂new,t ⋄ col[1]←⊂(⊂np),c ⋄ colI[1]←⊂(⊂np),cI
     lex[1]←⊂Last[1],gcol,(⊂np),(2↓(-M)↓lex[1]),(-⍳M)
     PC←⊂np ⋄ k←1 ⋄ →FPR

⍝*    Find next pivot row PR : uses k=1,PC
FPR: BF←0 ⋄ n←k-1 ⋄ t←k⊃T ⋄ r←k⊃row ⋄ c←k⊃col ⋄ lx←k⊃lex ⋄ y←t[;c⍳PC]
     R←((x←y>0)/r)~V0,v0 ⋄ i←(x/⍳⍴y)~r⍳V0,v0 ⋄ I←l←0 ⋄ →((0<⍴R),(n=0))/FR1,End ⋄ 1÷0

FR1: PR←R[1] ⋄ →((1≥⍴R)∨(⍴lx)<l←l+1)/Pvt ⋄ PR←I←lx[l] ⋄ →(I∊R)/Pvt ⍝←Select I if basic
     →(~I∊c)/FR1 ⋄ slx←¯1*I≠Last[k] ⋄ iR←x∊⌊/x←slx×t[i;c⍳I]÷y[i]
     R←iR/R ⋄ i←iR/i ⋄ →FR1  ⍝ Select rows with min ratio until only one row=PR

⍝**   Finish
Fin: Time←⎕ts-ts ⋄ Time←(30 24 60 60 1000⊥¯5↑Time)÷1000 ⋄ NoEq←NoEq+1 ⋄ s←0⍴0 ⋄ n←1
F1:  k←n+1 ⋄ s←s,((k⊃T)⍪0)[(k⊃row)⍳n⊃Le;(k⊃col)⍳Last[k]]÷D[k] ⋄ →(N≥n←n+1)/F1
     sigma←DAReport s ⋄ →((MultiEq≠1)∨Abort=1)/0 ⋄ →Mul ⍝ DAReport reports results

⍝**   Seek more equilibria, if MultiEq=1. ↓Reverse sign of lambda to continue path
Mul: t←1⊃T ⋄ t[;(1⊃col)⍳gcol]←-t[;(1⊃col)⍳gcol] ⋄ T[1]←⊂t
     ts←⎕ts ⋄ NoPivots←0 ⋄ PC←gcol ⋄ k←1 ⋄ →FPR

End: →(MultiEq=1)/0 ⋄ ⎕←'Err at FPR: no pivot row found' ⋄ 1÷0
```

Function PivotD – does pivoting, replicates Pvt in DA

```
k PivotD pcr;c;r;k;pr;prow;pc;pcol;pvt;sp;t;x
⍝∇ PivotD - Pivots pc,pr retaining integers

pc←pcr[1] ⋄ pr←pcr[2] ⋄ r←k⊃row ⋄ c←k⊃col
pr←r⍳pr ⋄ pc←c⍳pc ⋄ t←k⊃T ⋄ prow←t[pr;] ⋄ pcol←t[;pc] ⋄ sp←×pvt←prow[pc]
t←((t×pvt)-pcol∘.×prow)÷sp×D[k] ⋄ t[pr;]←sp×prow ⋄ t[;pc]←-sp×pcol
t[pr;pc]←sp×D[k] ⋄ x←r[pr] ⋄ r[pr]←c[pc] ⋄ c[pc]←x ⋄ row[k]←⊂r ⋄ col[k]←⊂c
T[k]←⊂t ⋄ D[k]←|pvt ⋄ NoPivots←NoPivots+1 ⋄ →0
```

★ ★ ★

Function Pay – recursive function to compute payoffs efficiently

```
Lv Pay x;Gn;k;n;xI
⍝∇ Pay - Recursive function for N-player payoffs at sigma=s.
⍝  Global:G,s,v;N,m;Le. Lv≡Level.

 →(Lv[1]≤0)/0 ⋄ →(N≥(⍴Lv)+n←Lv[1])/0 ⋄ →(N=⍴Lv)/Calc ⋄ k←1
Next: ((n-k),Lv) Pay s[(n-k)⊃xLe]∘.×x
 →((0<Lv[1]-k)∧(2≥k←k+1))/Next ⋄ →0
Calc: n←((~(⍳N)∊Lv)/⍳N)[1] ⋄ k←(1⌽⍳n),n+⍳(N-n) ⋄ xI←xIm ⋄ xI[n]←⊂⍳m[n]
 Gn←(m[n],×/xm[(⍳N)~n])⍴k⍉(n,xI)⌷G ⋄ v[n⊃Le]←Gn+.×,x ⋄ →0
```

★ ★ ★

Function Payoff – calls Pay

```
v ← Payoff s;n;x;xLe;xIm;xm;xs
⍝∇ Payoff - Computes payoffs from strategies s by calling on recursive function Pay

⍝ Limit calculations in <Pay> to support of s
    xLe←xIm←xm←⍬ ⋄ n←1
L2: xIm←xIm,⊂xs←(s[n⊃Le]≠0)/⍳m[n] ⋄ xm←xm,⍴xs ⋄ xLe←xLe,⊂(+/¯1↓xm)+⍳⍴xs
    →(N≥n←n+1)/L2 ⋄ s←s~0
  v←M⍴0 ⋄ (,N+1)Pay(,1) ⋄ →0 ⍝ Input s and Output v are global within Pay
```

Function Simplex – computes the current simplex from An, Seq, LS

```
V ← Simplex;i;j;jj;k;n;pn;q;r;ov;v
⍝∇ Simplex - Computes set V of vertices of 0's simplex of pure strategies
⍝∇           from Anchor,Sequence,LS
⍝ Global: An≡Anchor is a list of (enclosed integer) proposals to the players.
⍝ Global: Seq is matrix of rows (n,j) of pairs, indicating order of increments
⍝  to Anchor
⍝ Global: LS is an N-vector of last str. decremented for each player.
⍝  2=≡An, N=⍴An, ((1+r),N)=⍴V. An's n-th element is an enclosed integer
⍝    proposal to player n
⍝  span Inc[n] ≡ those str. of n incremented or decremented
⍝  LS[n] ≡ last str. decremented (=0 if none)

⍝ Compute support P and span Inc of simplex V defined by An,Seq,LS
    P←Inc←IncP←0⍴0 ⋄ n←1
L0: P←P,⊂(((n⊃An)>0)∨(⍳m[n])∊(Seq[1;]∊n)/Seq[2;])/⍳m[n]
    k←(Seq[1;]∊n)/Seq[2;] ⋄ j←k[⍋k] ⋄ →((0<⍴j)∧LS[n]≤⌈/j)/Flaw
    Inc←Inc,⊂q←j,LS[n] ⋄ IncP←IncP,⊂((k⍳q)<(k⍳¯1⌽q))/q
    →(N≥n←n+1)/L0

⍝ Compute the vertices of the simplex V
    r←⍴Seq ⋄ r←r[2] ⋄ V←(1,⍴ov)⍴ov←An ⋄ →(r=0)/0 ⋄ i←1
L1: v←ov ⋄ n←Seq[1;i] ⋄ j←Seq[2;i] ⋄ pn←n⊃v ⋄ q←n⊃Inc
    jj←q[1+q⍳j] ⋄ pn[j,jj]←pn[j,jj]+1,¯1 ⋄ v[n]←⊂pn ⋄ V←V⍪v ⋄ ov←v
    →(r≥i←i+1)/L1
```