Econometrica, Vol. 74, No. 2 (March, 2006), 397–429

HARD-TO-SOLVE BIMATRIX GAMES

BY RAHUL SAVANI AND BERNHARD VON STENGEL[1]

The Lemke–Howson algorithm is the classical method for finding one Nash equilibrium of a bimatrix game. This paper presents a class of square bimatrix games for which this algorithm takes, even in the best case, an exponential number of steps in the dimension d of the game. Using polytope theory, the games are constructed using pairs of dual cyclic polytopes with $2d$ suitably labeled facets in d-space. The construction is extended to nonsquare games where, in addition to exponentially long Lemke–Howson computations, finding an equilibrium by support enumeration takes on average exponential time.

KEYWORDS: Bimatrix game, computational complexity, Lemke–Howson algorithm, Nash equilibrium, support enumeration.

1. INTRODUCTION

THIS ARTICLE CONCERNS the computational problem of finding one Nash equilibrium of a bimatrix game, a two-player game in strategic form. Bimatrix games are among the most basic models in game theory. For applications of game theory, a computer program for finding Nash equilibria simplifies the task of analyzing a game. The GAMBIT project (McKelvey, McLennan, and Turocy (2005)) for solving games is intended to be such a research tool. When solving larger games, the running time of the computer program becomes important.

In computer science, the running time of a program is measured as a function of the size of the input; for the problem of finding a Nash equilibrium of a bimatrix game, this is the number of bits required to specify the payoff matrices. Problems that can be solved in *polynomial* running time are considered to be "computationally tractable" (see Garey and Johnson (1979), Papadimitriou (1994)).[2] At present, it is not known whether a Nash equilibrium of a bimatrix game can be found in polynomial time. This has been called one of the

[1] Rahul Savani is supported by an EPSRC doctoral grant. We thank the participants of the Bellairs Workshop on Polytopes, Games, and Matroids in Barbados in March 2003, in particular the organizer, Komei Fukuda, and Walter Morris and Jörg Rambau, for stimulating discussions. Andrew McLennan and Rabee Tourky noted an alternative proof of one of our results using imitation games. The comments of four referees and a co-editor helped to improve our paper significantly.

[2] Polynomial-time algorithms are even more important when computing power improves, since available computing time often determines the size of problems that are solved. As an example, consider two algorithms that take 2^n and n^3 many steps, respectively, for an input of size n. The first, exponential-time algorithm is superior to the second, polynomial-time algorithm when $n < 10$. With a very slow computer, where each step takes a minute, the practical limit of solving problems may be size $n = 10$, where both algorithms take about 1,000 steps and about 17 hours. With a 2^{16} times faster computer, where each step takes less than a millisecond, the exponential algorithm can solve problems of size 26 in about 17 hours, but the polynomial algorithm can solve problems of size 400 (and it solves the problem of size 26 in 17 seconds).

two “most important concrete open questions” in theoretical computer science (Papadimitriou (2001)).

A standard method for finding one Nash equilibrium of a bimatrix game is the algorithm developed by Lemke and Howson (1964), here called the LH algorithm. We present a class of games where this algorithm takes an exponential number of steps, which shows that the algorithm is not polynomial. The LH algorithm is a pivoting method related to the simplex algorithm for linear programming (Dantzig (1963)). Klee and Minty (1972) constructed linear programs for which the simplex method with a certain pivot rule takes an exponential number of pivoting steps; similar examples have been constructed for other pivot rules (see Klee and Kleinschmidt (1987), Todd (2001)). The LH algorithm has no choice of its pivot rule, but free choice, corresponding to a pure strategy of one of the players, of its first step (the first variable “to enter the basis”). In the games constructed here, the running time of the LH algorithm is exponential even for the *best* first choice of the algorithm. To our knowledge, these are the first examples of this kind. Finding a Nash equilibrium in subexponential time must therefore go beyond this classic pivoting approach.

When the game is zero-sum, an equilibrium is the solution to a linear program. Linear programs can be solved in polynomial time by interior point methods (see Todd (2001)). No interior point method is known for finding Nash equilibria. One difficulty is that the set of Nash equilibria of a bimatrix game is generally not convex. Problems relating to that set tend to be computationally difficult; for example, the problem of deciding whether a game has a unique Nash equilibrium (Gilboa and Zemel (1989), Conitzer and Sandholm (2003), Codenotti and Štefankovič (2005)). Finding all equilibria is therefore computationally intractable for larger games. However, it is conceivable that one Nash equilibrium can always be found in polynomial time.

Our construction uses the theory of polyhedra (see Ziegler (1995) or Grünbaum (2003)). This geometric view gives a good insight into the structure of Nash equilibria of two-player games (see von Stengel (2002)). The set of each player’s mixed strategies together with the best response payoff to the other player is described by a polyhedron. A vertex of each polyhedron is obtained by converting some inequalities into equations, which describe the support of a mixed strategy and its best responses. An equilibrium is given by a “complementary” vertex pair. The LH algorithm traverses “almost complementary” edges of the polyhedra until it reaches an equilibrium.

We use a standard construction of “dual cyclic polytopes.” These are polyhedra for which the vertex-defining inequalities are known in arbitrary dimension (Gale (1963)). This purely combinatorial information can also be used to construct bimatrix games with many equilibria (von Stengel (1999)).

Our work is most closely related to Morris (1994), who used dual cyclic polytopes to produce exponentially long “Lemke paths” on polytopes. These paths correspond to a “symmetric” version of the LH algorithm, which finds symmetric equilibria of a symmetric game. The games obtained from Morris’s construction have additional nonsymmetric equilibria. The standard LH algorithm always terminates, very quickly, at such a nonsymmetric equilibrium. These games are therefore not directly useful for our purpose. After we found our construction of long LH paths for bimatrix games, McLennan and Tourky (2005) noted an ingenious alternative proof of exponentially long LH paths based on Morris's construction and *imitation games*, as we explain in Section 5.

As described so far, we construct square games that have a unique equilibrium, which is completely mixed. The games derived from Morris (1994) by McLennan and Tourky (2005) also have a unique, completely mixed equilibrium. The LH algorithm needs exponential time to find it. However, the equilibrium is quickly found by a simple algorithm called *support enumeration* (e.g., Dickhaut and Kaplan (1991), Porter, Nudelman, and Shoham (2004), or Bárány, Vempala, and Vetta (2005)). The support of a mixed strategy is the set of pure strategies that it plays with positive probability. All pure strategies in the support of an equilibrium strategy must have equal expected payoff. If, as here, the game is nondegenerate, the corresponding linear equations uniquely determine the mixed strategy probabilities of the other player. Support enumeration considers all possible supports for both players and the solutions to the respective linear equations, and checks if these solutions define an equilibrium. In a square game, it is natural to test the set of all pure strategies as an equilibrium support, which gives the equilibrium in our games. That is, our square games are not "hard to solve" by other methods. As an additional result, we extend our construction to nonsquare games where *both* the LH algorithm and, on average, support enumeration are exponential.

In Section 2, we describe a modification of the LH algorithm that finds a symmetric equilibrium of a symmetric bimatrix game. We explain this modified and simpler LH method because, to our knowledge, it has not been described earlier (although it is straightforward) and because it leads naturally to the usual LH method.

Section 3 gives our construction of square games. The LH paths are defined purely combinatorially in terms of the supports of and best responses to the mixed strategies that they trace. These correspond to known binary patterns that encode the vertices of dual cyclic polytopes. Linear recurrences for the various path lengths give rise to their exponential growth. For $d = 2, 4, 6, \ldots$, the length of the longest path for a $d \times d$ game is given by every third Fibonacci number, which is proportional to $\phi^{3d/2}$, where $\phi = 1.618\ldots$ is the Golden Ratio. Shorter path lengths are obtained by certain sums of these numbers, the shortest length being proportional to $\phi^{3d/4}$.

Section 4 describes how to extend the construction to nonsquare games, with equilibria that are hard to find by support enumeration, as well as by the LH algorithm.

Section 5 explains how to interpret the Lemke paths of Morris (1994) as paths of the symmetric LH algorithm described in Section 2. All ordinary LH paths for these symmetric games are very short and lead to nonsymmetric pure-strategy equilibria. We then explain the use of imitation games in this context, as suggested by McLennan and Tourky (2005).

In Section 6, we discuss related work on the computational complexity of linear complementarity problems, which generalize the game equilibrium problem. We mention open questions that arise in that context and give an example of a game that results from our construction.

2. GAMES, POLYTOPES, AND THE LEMKE–HOWSON ALGORITHM

We use the following notation. Given a bimatrix game (A, B) with $m \times n$ payoff matrices A and B, a mixed strategy for player 1 is a vector x in $\mathbb{R}^m$ with nonnegative components that sum to 1. A mixed strategy for player 2 is such a vector y in $\mathbb{R}^n$. All vectors are column vectors; the row vector that corresponds to x is written as the transpose $x^\top$. The support of a mixed strategy is the set of pure strategies that have positive probability. A best response to y is a mixed strategy x of player 1 that maximizes his expected payoff $x^\top Ay$, and a best response to x is a mixed strategy y of player 2 that maximizes her expected payoff $x^\top By$. A Nash equilibrium is a pair of mutual best responses. Best responses are characterized by the following combinatorial condition, which we state only for a mixed strategy x of player 1.

LEMMA 1 (Nash (1951)): *Let x and y be mixed strategies of players* 1 *and* 2, *respectively. Then x is a best response to y if and only if all strategies in the support of x are pure best responses to y.*

A game (A, B) is *symmetric* if $A = B^\top$, so it does not change when the players change roles. The game of "chicken" with $A = B^\top = \left(\begin{smallmatrix} 2 & 2 \\ 4 & 1 \end{smallmatrix}\right)$ is an example. Its equilibria, in terms of probability vectors, are the bottom left pure strategy pair $((0, 1)^\top, (1, 0)^\top)$ with payoffs $4, 2$ to players 1 and 2, respectively, the top right pure strategy pair $((1, 0)^\top, (0, 1)^\top)$ with payoffs $2, 4$, and the mixed strategy pair $((1/3, 2/3)^\top, (1/3, 2/3)^\top)$ with payoffs $2, 2$. The mixed strategy equilibrium is the only symmetric equilibrium. Its probabilities are uniquely determined by the condition that the pure strategies in the support of the opponent's strategy must both be best responses (by Lemma 1) and hence have equal expected payoff.

In a mixed equilibrium, the probabilities are uniquely given by the pair of supports if the corresponding submatrices have full rank; the support sizes are then equal. This holds if the game is *nondegenerate*, defined by the property that the number of pure best responses to any mixed strategy never exceeds the size of its support (see von Stengel (2002) for a detailed discussion). The LH algorithm can be extended to degenerate games by standard lexicographic perturbation techniques. All games considered here are nondegenerate.

By Lemma 1, an equilibrium is given if any pure strategy of a player is either a best response (to the opponent's mixed strategy) or is played with probability zero (by the player himself). This can be captured by polytopes (see Ziegler (1995), Grünbaum (2003)) whose facets represent pure strategies, either as best responses or having probability zero. We explain first the simpler case of symmetric equilibria of a symmetric game with $d \times d$ payoff matrix C to player 1, say. We then extend this easily to nonsymmetric games. Let

$$S = \{z \in \mathbb{R}^d \mid z \geq \mathbf{0}, Cz \leq \mathbf{1}\}, \tag{1}$$

where $\mathbf{0}$ and $\mathbf{1}$ denote vectors with all entries 0 and 1, respectively, and inequalities hold for all components. We can assume that *C is nonnegative and has no zero column* by adding a constant to all payoffs, which does not change the best response structure, so that the polyhedron S is bounded and thus a polytope. We assume there are no redundant inequalities in $Cz \leq \mathbf{1}$, which would correspond to dominated strategies. Then the game is nondegenerate if and only if the polytope S is *simple*, that is, every vertex lies on exactly d facets of the polytope. A facet is obtained by making one of the inequalities defining the polytope *binding*, that is, by converting it into an equality. The following lemma characterizes Nash equilibria in terms of polytope vertices, as already shown by Vorob'ev (1958).

LEMMA 2: *A mixed strategy pair (x, y) is a symmetric Nash equilibrium of the game $(C, C^\top)$ if and only if $x = y = u \cdot z$ and $z \in S$ in* (1), *$z \neq \mathbf{0}$, $u = 1/\sum_i z_i$, and $z^\top(\mathbf{1} - Cz) = 0$, where z must be a vertex of S by nondegeneracy.*

PROOF: Let $z \in S$, $z \neq \mathbf{0}$, and $u = 1/\sum_i z_i$. Then $u > 0$ and zu is a mixed strategy x. The condition $Cz \leq \mathbf{1}$ is equivalent to $Cx \leq \mathbf{1}u$. The orthogonality condition $z^\top(\mathbf{1} - Cz) = 0$ is equivalent to $x^\top(\mathbf{1}u - Cx) = 0$, so that for each positive component x_i of x (of which there is at least one), $(Cx)_i = u = \max_k(Cx)_k$. Thus, by Lemma 1, x is a best response to itself, that is, (x, x) is a symmetric equilibrium. Conversely, any such equilibrium (x, x), with $u = \max_k(Cx)_k > 0$ and $z = x \cdot 1/u$, gives a vector z with the stated properties.

The vector z is on d facets of S since for each i, either $z_i = 0$ or $(Cz)_i = 1$. If z was not a vertex but on a higher-dimensional face of S, any vertex of that face would be on additional facets, contradicting nondegeneracy of the game because S would then not be a simple polytope. *Q.E.D.*

In the game of chicken above, $z = (1/6, 1/3)^\top$ gives the symmetric equilibrium. The vector z has to be rescaled to become a mixed strategy x. The equilibrium payoff u, normalized to 1 in $Cz \leq \mathbf{1}$, is the scaling factor. The converse mapping from x to z defines a projective transformation of a polyhedron that represents the "upper envelope" of expected payoffs to the polytope S (see von Stengel (2002)).

The conditions in Lemma 2 define a *linear complementarity problem* (LCP) (see Cottle, Pang, and Stone (1992)), usually stated as follows: Find z so that

$$(2) \qquad z \geq \mathbf{0}, \quad q + Mz \geq \mathbf{0}, \quad z^\top(q + Mz) = 0,$$

here with data $M = -C$, $q = \mathbf{1}$. This LCP has a trivial solution $z = \mathbf{0}$, which is not a Nash equilibrium. However, $\mathbf{0}$ is an *artificial equilibrium*, which is the starting point of what we call the *symmetric LH algorithm*.

Given a nondegenerate symmetric game $(C, C^\top)$, the symmetric LH algorithm finds a nonzero vertex z of the polytope S in (1) so that $z^\top(\mathbf{1} - Cz) = 0$, giving a symmetric Nash equilibrium by Lemma 2.

It is useful to *label* the facets of S, as done by Shapley (1974). For each pure strategy i, the facets defined by $z_i = 0$ and by $(Cz)_i = 1$ both get label i. Every vertex has the label of the facets on which it lies. The complementarity condition $z^\top(\mathbf{1} - Cz) = 0$ then means that z is *completely labeled* (has all labels i), since then either $z_i = 0$ or $(Cz)_i = 1$ (or both, but this cannot occur because S is simple, so a completely labeled vertex has each label exactly once).

The LH algorithm is started from the completely labeled vertex $z = \mathbf{0}$ by choosing one label k that is initially *dropped*, meaning that label k is no longer required. This is the only free choice of the algorithm, which from then on proceeds in a unique manner. By leaving the facet with label k, a unique edge is traversed whose endpoint is another vertex, which lies on a new facet. The label, say j, of that facet, is said to be *picked up*. If this is the *missing* label k, the algorithm terminates at a completely labeled vertex. Otherwise, j is clearly *duplicate* and the next edge is (uniquely) chosen by leaving the facet that so far had label j, and the process is repeated. The LH method generates a sequence of k-*almost complementary* edges and vertices (having all labels except possibly k, where k occurs only at the starting point and endpoint). The resulting path cannot repeat a vertex because as this would offer a second way to proceed when that vertex is first encountered, which is not the case (since S is simple). Hence, it terminates at a Nash equilibrium.

As in the simplex algorithm (Dantzig (1963)), edge traversal is implemented algebraically by *pivoting* with variables entering and leaving a basis, the nonbasic variables representing the current facets. The only difference is the rule for choosing the next entering variable, which in the simplex algorithm depends on the objective function. Here, the *complementary pivoting* rule chooses the nonbasic variable with duplicate label to enter the basis.

For nonsymmetric bimatrix games (A, B) or even for finding nonsymmetric equilibria of symmetric games as in the game of "chicken" above, the LH algorithm is applied as follows, which is its standard form. Let

$$(3) \qquad z = \begin{pmatrix} x \\ y \end{pmatrix}, \quad C = \begin{pmatrix} 0 & A \\ B^\top & 0 \end{pmatrix}.$$

The polytope S of dimension $d = m + n$ in (1) is then the *product* $P \times Q$ of the polytopes

$$(4) \qquad P = \{x \in \mathbb{R}^m \mid x \geq \mathbf{0}, B^\top x \leq \mathbf{1}\}, \quad Q = \{y \in \mathbb{R}^n \mid Ay \leq \mathbf{1}, y \geq \mathbf{0}\}.$$

Any Nash equilibrium (x, y) of (A, B) is again given by $z^\top(\mathbf{1} - Cz) = 0$, which is equivalent to $x^\top(\mathbf{1} - Ay) = 0$ and $y^\top(\mathbf{1} - B^\top x) = 0$. These conditions state that x is a best response to y and vice versa, where x and y have to be normalized to represent mixed strategies. The only difference from Lemma 2 is that this normalization has to be done separately for x and y, rather than for the entire vector z. It is easy to see that in equilibrium $x = \mathbf{0}$ if and only if $y = \mathbf{0}$, and then $(\mathbf{0}, \mathbf{0})$ is the artificial equilibrium.

The LH algorithm is applied as before, where a label corresponds either to a strategy i of player 1 or a strategy j of player 2. These have to be distinct, so it is convenient to number the n strategies of player 2 as $m + 1, \ldots, m + n$, as suggested by Shapley (1974). A label then represents a pure strategy that has probability zero or is a best response. A label i with $1 \leq i \leq m$ is a strategy of player 1 and determines the facet $x_i = 0$ of P or $(Ay)_i = 1$ of Q, corresponding to the respective ith inequality in (4) that becomes binding. A label j with $m + 1 \leq j \leq m + n$ is a strategy of player 2 and determines the facet $(B^\top x)_j = 1$ of P or $y_j = 0$ of Q, which is the respective jth binding inequality in (4).

The LH path using the edges of $S = P \times Q$ is a subgraph of the *product graph* of the edge graphs of P and Q. This means that edges are alternately traversed in P and Q, keeping the vertex in the other polytope fixed. A duplicate label picked up in P is dropped in Q and vice versa. This is the standard view of the LH algorithm; for further details see von Stengel (2002).

3. LEMKE–HOWSON ON LABELED DUAL CYCLIC POLYTOPES

We construct games by defining P and Q in (4) as the well-understood "dual cyclic polytopes" (see Ziegler (1995) or Grünbaum (2003)), similar to von Stengel (1999). These polytopes are in dimension d and have f facets, where $d = m$ for P and $d = n$ for Q, and in both cases $f = m + n$.

A standard way to obtain a *cyclic polytope* P' in dimension d with f vertices is to take the convex hull of f points $\mu(t_i)$ on the *moment curve* $\mu : t \mapsto (t, t^2, \ldots, t^d)^\top$ for $1 \leq i \leq f$. However, the polytopes in (4) are defined by inequalities and not as convex hulls of points. In the *dual* of a polytope, its vertices are reinterpreted as normal vectors of facets. The polytope P' is first translated so that it has the origin $\mathbf{0}$ in its interior, for example by subtracting the arithmetic mean $\overline{\mu}$ of the points $\mu(t_i)$ from each such point. The resulting vectors $c_i = \mu(t_i) - \overline{\mu}$ then define the *dual cyclic polytope*

$$P'' = \{z \in \mathbb{R}^d \mid c_i^\top z \leq 1, 1 \leq i \leq f\}.$$

Both P and Q in (4) will be dual cyclic polytopes with a special order of their inequalities corresponding to the facet labels. A suitable affine transformation, described in von Stengel (1999, p. 560) or Savani and von Stengel (2004, Appendix A), gives P from P'' and gives Q in a similar manner, so that the first m inequalities (for the pure strategies of player 1) in P have the form $x \geq \mathbf{0}$ and the last n inequalities (for the pure strategies of player 2) in Q are $y \geq \mathbf{0}$. The last n inequalities $B^\top x \leq \mathbf{1}$ in P and the first m inequalities $Ay \leq \mathbf{1}$ in Q then determine the game (A, B). The game data are of polynomial size in $m+n$, so the running time of an algorithm with the game as input is polynomial if and only if it is polynomial in $m+n$.

A vertex u of a dual cyclic polytope in dimension d with f facets is characterized by the *bit string* $u_1 u_2 \cdots u_f$ of length f, with the kth bit u_k indicating whether u is on the kth facet ($u_k = 1$) or not ($u_k = 0$). The polytope is simple, so exactly d bits are 1; the other $f - d$ bits are 0. Assume that $t_1 < t_2 < \cdots < t_f$ when defining the kth facet of P'' by the binding inequality $(\mu(t_k) - \overline{\mu})z \leq 1$. Then the vertices of P'' are characterized by the bit strings that fulfill the *Gale evenness* condition (Gale (1963)): A bit string represents a vertex if and only if any substring of the form $01 \cdots 10$ has even length, so 0110, 011110, etc. is allowed, but not 010, 01110, and so on. A maximal substring of 1's is called a *run*. We consider only *even* dimensions d, where the allowed odd runs of 1's at both ends of the string can be glued together to form an even run, which shows the cyclic symmetry of the Gale evenness condition. Let $G(d, f)$ be the set of these bit strings of length f with d 1's fulfilling Gale evenness.

For the rest of the paper, both m and n are even, and $m \leq n$. The vertices of P and Q are described by the sets of bit strings $G(m, m+n)$ and $G(n, m+n)$, respectively. The 1's in a bit string encode the facets to which the vertex belongs. We also need *facet labels* for the complementarity condition and the LH algorithm. The facet labels are defined by permutations l and l' of $1, \ldots, m+n$ for P and Q, respectively. For a vertex u of P, which we identify with its bit string in $G(m, m+n)$, its set of labels is given by $\{l(k) \mid u_k = 1, 1 \leq k \leq m+n\}$. The kth facet of P (corresponding to the kth position in a bit string) has label $l(k) = k$, so l is simply the identity permutation. A vertex v of Q is identified with a bit string in $G(n, m+n)$ and its set of labels is $\{l'(k) \mid v_k = 1, 1 \leq k \leq m+n\}$. The kth facet of Q has label $l'(k)$. The permutation l' has the fixed points $l'(1) = 1$ and $l'(m) = m$, and otherwise exchanges adjacent numbers, as follows:

$$(5) \qquad l'(k) = \begin{cases} k, & k = 1, m, \\ k + (-1)^k, & 2 \leq k \leq m-1, \\ k - (-1)^k, & m+1 \leq k \leq m+n. \end{cases}$$

Let $\Gamma(m, n)$ be a game defined in this way.

The artificial equilibrium e_0 of $\Gamma(m, n)$ is a vertex pair (u, v) so that u is labeled with $1, \ldots, m$ and v is labeled with $m+1, \ldots, m+n$, so that we have complementarity. In terms of bit strings, $u = 1^m 0^n$ (which are m 1's followed by n 0's) and $v = 0^m 1^n$, which both fulfill Gale evenness, and have the indicated labels under l and l', respectively, so that

$$e_0 = (1^m 0^n, 0^m 1^n) \in G(m, m+n) \times G(n, m+n). \tag{6}$$

The following lemma states that in any Nash equilibrium of $\Gamma(m, n)$, player 1's strategy has full support.

LEMMA 3: *Consider a Nash equilibrium of $\Gamma(m, n)$, represented by a pair of bit strings (u, v) in $G(m, m+n) \times G(n, m+n)$. Then $u = 0^m s$ and $v = 1^m t$ for some bit strings s and t of length n.*

PROOF: The vertex pair (u, v) is completely labeled and is not the artificial equilibrium e_0. Either $u_m = 1$ or $v_m = 1$. We begin with the latter case, so $u_m = 0$. If $v_{m+1} = 1$, then $u_{m+2} = 0$ (via complementarity, since $l'(m+1) = m+2$), so $u_{m+1} = 0$ by Gale evenness and thus $v_{m+2} = 1$. Continuing in that way, all 1's to the right of the mth bit v_m of v (which is 1) have to come in pairs. Similarly, if $v_{m-1} = 1$, then $u_{m-2} = 0$ by complementarity, which with $u_m = 0$ implies $u_{m-1} = 0$ and $v_{m-2} = 1$. This means that the 1's to the left of v_m come in pairs if there is a zero to the left of them. However, then the run of 1's that contains v_m has odd length and must include v_{m+n}, and then it is too long. Hence, there is no zero in v to the left of v_m, and $v = 1^m t$ and $u = 0^m s$ for some bit strings s and t of length n, as claimed.

In the same way, $u_m = 1$ implies that all bits in u to the left of u_m are 1 and, since u has only m bits equal to 1, all bits to the right of u_m are 0, so that $(u, v) = e_0$, which is the artificial equilibrium and is not a Nash equilibrium. *Q.E.D.*

COROLLARY 4: *The only equilibrium of the game $\Gamma(d, d)$ is $e_1 = (0^d 1^d, 1^d 0^d)$.*

PROOF: For $m = n = d$, the vertices u and v in Lemma 3 are bit strings that contain d 1's, so that $s = 1^d$ and $t = 0^d$. *Q.E.D.*

In the remainder of this section, we consider only square games where $d = m = n$. Then, by Corollary 4, all LH paths for any missing label lead from e_0 to e_1. We analyze these paths for square games. It will then be easy to describe the LH paths for the nonsquare games $\Gamma(d, 2d)$, which are treated in Section 4.

Denote by $\pi(d, k)$ the LH path with missing label k for the game $\Gamma(d, d)$. We regard $\pi(d, k)$ as a sequence $(u^0, v^0)(u^1, v^1) \cdots (u^L, v^L)$ of vertex pairs in $P \times Q$, that is, in $G(d) \times G(d)$, where $G(d)$ abbreviates $G(d, 2d)$. Let $L(d, k) = L$ be the length of that path.

As an example, Figure 1 shows $\pi(4, 1)$. The numbers at the top are the labels $l(k)$ and $l'(k)$ for $k = 1, \ldots, 8$. The vertices of P and Q are shown as bit patterns, where, for better visual distinction of the bits, a zero bit is written as a dot. The 20 steps of this path are indicated at the side, where the odd-

step	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
	𝟙	𝟙	𝟙	𝟙	·	·	·	·
1 ⟨	·	𝟙	𝟙	𝟙	𝟙	·	·	·
3 ⟨	·	𝟙	𝟙	·	𝟙	𝟙	·	·
5 ⟨	·	·	𝟙	𝟙	𝟙	𝟙	·	·
7 ⟨	·	·	·	𝟙	𝟙	𝟙	𝟙	·
9 ⟨	·	·	𝟙	𝟙	·	𝟙	𝟙	·
11 ⟨	·	𝟙	𝟙	·	·	𝟙	𝟙	·
13 ⟨	·	𝟙	𝟙	·	·	·	𝟙	𝟙
15 ⟨	·	·	𝟙	𝟙	·	·	𝟙	𝟙
17 ⟨	·	·	·	𝟙	𝟙	·	𝟙	𝟙
19 ⟨	·	·	·	·	𝟙	𝟙	𝟙	𝟙

1	**3**	**2**	**4**	**6**	**5**	**8**	**7**	step
·	·	·	·	𝟙	𝟙	𝟙	𝟙	
·	·	·	𝟙	𝟙	·	𝟙	𝟙	⟩ 2
·	·	𝟙	𝟙	·	·	𝟙	𝟙	⟩ 4
·	𝟙	𝟙	·	·	·	𝟙	𝟙	⟩ 6
·	𝟙	𝟙	·	·	𝟙	𝟙	·	⟩ 8
·	·	𝟙	𝟙	·	𝟙	𝟙	·	⟩ 10
·	·	·	𝟙	𝟙	𝟙	𝟙	·	⟩ 12
·	·	𝟙	𝟙	𝟙	𝟙	·	·	⟩ 14
·	𝟙	𝟙	·	𝟙	𝟙	·	·	⟩ 16
·	𝟙	𝟙	𝟙	𝟙	·	·	·	⟩ 18
𝟙	𝟙	𝟙	𝟙	·	·	·	·	⟩ 20

FIGURE 1.—The path $\pi(4,1)$ with vertices of P and Q as bit strings. A dot represents a zero bit.

numbered steps change the vertex in P and the even-numbered steps change the vertex in Q. Step i changes the vertex pair (u^{i-1}, v^{i-1}) to (u^i, v^i). The starting point e_0 is the vertex pair $e_0 = (u^0, v^0) = (11110000, 00001111)$. Step 1 is to drop label 1 in P from u^0, so the bit u_1^0 changes from 1 to 0. By Gale evenness, this gives the bit string 01111000 as the new vertex u^1 in P. In Figure 1, the bit 1 that is changed to 0 has a little downward arrow "v" underneath it, with the new bit that changes from 0 to 1 indicated with that arrow above the new bit 1 in the next vertex. In u^1, label 5 has been picked up, which is now duplicate and dropped from vertex v^1 in Q (where $v^1 = v^0$), giving the next vertex $v^2 = 00011011$ in step 2. In P, the vertex u^2 is unchanged, $u^2 = u^1$. The new duplicate label is 4. Hence, in step 3, label 4 is dropped in P, giving vertex $u^3 = 01101100$. In that manner, the path proceeds until it ends at $(u^{20}, v^{20}) = e_1$.

We will show that all paths can be expressed in terms of the two special paths $\pi(d,1)$ and $\pi(d,2d)$. These have certain symmetries. Figure 1 illustrates the symmetry of $\pi(d,1)$, which is stated in the next lemma, for $d = 4$.

LEMMA 5: *Let $L = L(d,1)$ and let (u^i, v^i) be the ith vertex pair of the path $\pi(d,1)$. Then for* $0 \le i \le L$, $(u^i, v^i) = (v^{L-i}, u^{L-i})$.

PROOF: The particular names of the labels do not matter, so we can rename them for both P and Q with the permutation l' in (5): the kth facet in P gets label $l'(l(k))$, which is $l'(k)$, and the kth facet in Q gets label $l'(l'(k))$, which is $l(k)$. Then P and Q switch roles, e_0 is exchanged with e_1, label 1 stays the same, and the path backwards corresponds to $\pi(d,1)$ itself, as claimed. *Q.E.D.*

The symmetry of the path $\pi(d,2d)$ is less easy to state. Figure 2 shows this path for $d = 6$. In that picture, disregard the last vertex in P, and the first and

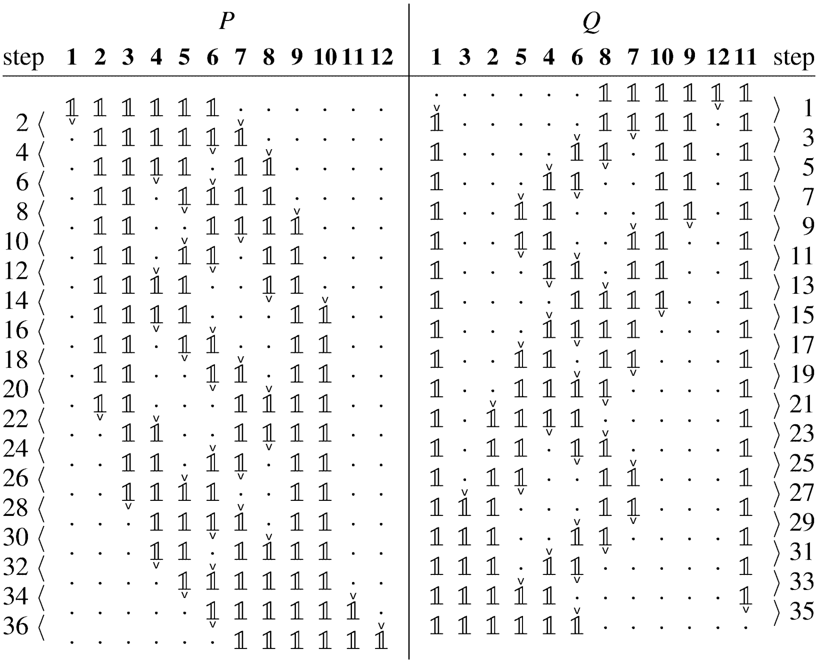

FIGURE 2.—The path $\pi(6,12)$.

last vertex in Q. Then the column labeled 12 in both P and Q has only 0's, because 12 is the missing label. When this column in both P and Q is also disregarded, the bit pattern of the path shows a symmetry in each polytope by "pointwise reflection," where the point of reflection is in column 6 in step 18 of P, and at the vertex that stays fixed in Q during that step. The pointwise symmetry means that in each polytope, writing each bit string backwards, while ignoring the bit corresponding to the missing label, gives the path in reverse direction (disregarding the first and the last two vertex pairs). For general d, this symmetry is stated in the next lemma, which we prove in the Appendix.

LEMMA 6: *Let* $L = L(d, 2d)$ *and let* (u^i, v^i) *be the ith vertex pair of the path* $\pi(d, 2d)$ *for* $0 \le i \le L$. *Let* $B(d)$ *be the subpath* $(u^1, v^1) \cdots (u^{L-2}, v^{L-2})$ *of* $\pi(d, 2d)$. *Then for the vertex pairs of* $B(d)$, *for* $1 \le i \le L-2$,

$$u_k^i = u_{2d-k}^{L-1-i}, \quad 1 \le k \le 2d-1, \tag{7}$$

$$v_1^i = v_{2d}^{L-1-i} = 1, \tag{8}$$

$$v_k^i = v_{2d-k}^{L-1-i}, \quad 2 \le k \le 2d-2, \tag{9}$$

$$u_{2d}^i = v_{2d-1}^i = 0. \tag{10}$$

In (u^1, v^1), *the duplicate label is* 1, *which is then dropped in P and never picked up again.*

The subpath $B(d)$ of $\pi(d, 2d)$, defined in Lemma 6, and the path $\pi(d, 1)$, which we call $A(d)$, are "building blocks" for other such paths in higher dimension by inserting constant bits in suitable positions. This is stated in the following central theorem. Two paths π and π' are concatenated by the following special path composition, which we denote by $\pi + \pi'$. Here, both π and π' are paths on $P \times Q = G(d) \times G(d)$. Let (u, v) be the last vertex pair of π and let (u', v') be the first vertex pair of π'. Then $\pi + \pi'$ is defined if (u, v) and (u', v') are joined by an edge in $P \times Q$, that is, either $u = u'$ and v is joined to v' by an edge of Q or $v = v'$ and u is joined to u' by an edge of P. The length of the new path $\pi + \pi'$ is the sum of the lengths of π and π' plus 1; the number of its *vertex pairs* is simply the respective sum.

THEOREM 7: *Let* $A(d) = \pi(d, 1)$ *and let* $B(d)$ *be as in Lemma* 6. *Then there are paths* $C(d)$ *and mappings* α, β, β', γ, γ' *defined on vertex pairs, and extended to sequences of vertex pairs, so that*

$$A(d) = \beta(B(d)) + C(d), \tag{11}$$

$$C(d) = \alpha(A(d-2)) + \beta'(B(d)), \tag{12}$$

$$B(d) = \gamma(A(d-2)) + \gamma'(C(d-2)). \tag{13}$$

The proof of Theorem 7 is given in the Appendix. We illustrate equation (11) for $d = 6$ using Figure 3, which shows the path $A(6)$. Comparing Figures 2 and 3, we see that steps 1–33 of $A(6)$ look almost like steps 2–34 of $\pi(6, 12)$, which are the 33 steps of $B(6)$. The only difference is that in $B(6)$, the bit in Q in column 1 is 1 and the bit in the column with label 12 is 0, whereas in $A(6)$ it is the other way around. The replacement of these two bits is performed by the mapping β, which is defined in (30) in the Appendix. The path $C(d)$ in (11) is simply a tail segment of $A(d)$. In Figure 3, $C(6)$ consists of steps 35–88 of $A(6)$. Step 34 of $A(6)$ is the edge in Q that joins the paths $\beta(B(6))$ and $C(6)$, represented by the "+" sign in (11) as defined before Theorem 7.

To illustrate equation (12), note that $C(6)$, beginning with step 35 of $A(6)$, starts like $\pi(4, 1)$ shown in Figure 1, which is $A(4)$. The bit strings in $A(4)$ are written backwards and extended by inserting a 0 bit at the front and adding the bits 110 at the end of the bit string in each polytope. This is done by the mapping α, which is defined in (31) in the Appendix, as are the other mappings in (12) and (13).

Let a_n be the number of vertex pairs of $A(2n)$, which is 1 more than the length $L(2n, 1)$ of that path. Let b_n and c_n be the number of vertex pairs of $B(2n)$ and $C(2n)$, respectively. That is,

$$a_n = L(2n, 1) + 1, \quad b_n = L(2n, 4n) - 2, \quad n \geq 1. \tag{14}$$

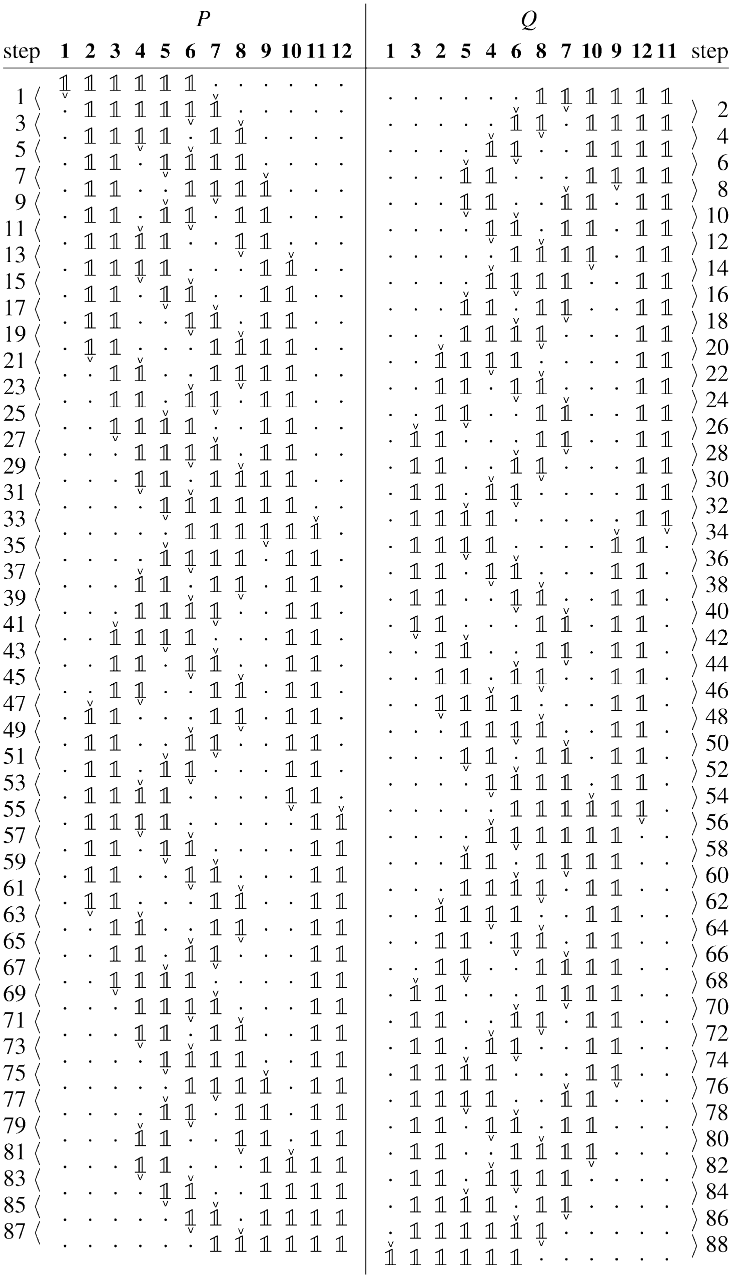
P
Q
step 1 2 3 4 5 6 7 8 9 10 11 12
1 3 2 5 4 6 8 7 10 9 12 11 step

FIGURE 3.—The path $A(6) = \pi(6, 1)$.

Then the concatenation of paths in (11) implies $a_n = b_n + c_n$, in (12) implies $c_n = a_{n-1} + b_n$, and in (13) implies $b_n = a_{n-1} + c_{n-1}$. Moreover, the paths $\pi(2, 1)$ and $\pi(2, 4)$ have length $4 = a_1 - 1 = b_1 + 2$. This shows that the numbers $b_1, c_1, a_1, b_2, c_2, a_2, \ldots$ are the Fibonacci numbers $2, 3, 5, 8, 13, 21, \ldots$ given by

$$(15) \qquad f_0 = 1, \quad f_1 = 2, \quad f_{n+1} = f_n + f_{n-1}, \quad n \geq 1,$$

that is,

$$(16) \qquad a_n = f_{3n}, \quad b_n = f_{3n-2}, \quad n \geq 1.$$

So both the lengths of $\pi(d, 1)$ and of $\pi(d, 2d)$ for even $d = 2n = 2, 4, 6, \ldots$ are given by every third Fibonacci number (minus 1 and plus 2, respectively). These are the longest paths. They occur several times, since, as shown next, $L(d, 1) = L(d, d)$ and $L(d, d+1) = L(d, d+2) = L(d, 2d-1) = L(d, 2d)$. This is due to the symmetry of the Gale evenness condition and of the labelings. Other paths $\pi(d, k)$ are given as concatenations of these paths in lower dimension. They are characterized, for all possible missing labels k, in the following theorem.

THEOREM 8: *The LH path lengths for any missing label are characterized by* (14)–(16), *and*

(a) $L(d, k) = L(d, d + 1 - k)$ *and* $L(d, d + k) = L(d, 2d + 1 - k)$ *for* $1 \leq k \leq d$;

(b) $L(d, k) = L(d, k + 1)$ *for even* k *when* $2 \leq k \leq d - 2$ *and for odd* k *when* $d + 1 \leq k \leq 2d - 1$;

(c) $L(d, k) = L(k, 1) + L(d - k, 1)$ *for even* k *and* $2 \leq k \leq d - 2$;

(d) $L(d, d+k) = L(k, 2k) + L(d-k+2, 2(d-k+2)) - 4 = b_{k/2} + b_{d/2-k/2+1}$ *when* k *is even and* $2 \leq k \leq d - 2$.

The proof of this theorem, which is similar to the proofs of Lemma 6 and Theorem 7, is given in the Appendix. Examples of paths where Theorem 8 applies can be found in Savani and von Stengel (2004). Using (b), cases (c) and (d) cover all possible missing labels.

It is easy to see that the shortest path lengths are obtained as follows: If d is divisible by 4, that is, $d/2$ is even, then the shortest path length occurs for missing label $d/2$ and is given by $L(d, d/2) = 2a_{d/2} - 2$ according to Theorem 8(c). If $d/2$ is odd, then the shortest path length occurs for missing label $3d/2$, where $L(d, 3d/2) = L(d, 3d/2 + 1) = 2b_{d/2+1}$ by Theorem 8(b) and (d). When $d/2$ is even, the path when dropping label $3d/2$ is only two steps longer than when dropping label $d/2$, because then $L(d, 3d/2) = b_{d/2} + b_{d/2+1} = b_{d/2} + a_{d/2} + c_{d/2} = 2a_{d/2}$. Therefore, the shortest path results essentially when dropping label $3d/2$.

The Fibonacci numbers (15) have the well-known explicit expression (see, for example, Graham, Knuth, and Patashnik (1994))

$$f_n = K\phi^n + \overline{K}\overline{\phi}^n, \quad \phi, \overline{\phi} = 0.5 \pm 0.5\sqrt{5}, \quad K, \overline{K} = 0.5 \pm 0.3\sqrt{5},$$

where $\phi = 1.618\ldots$ is the Golden Ratio and $K = 1.170\ldots$. Then f_n is $K\phi^n$ rounded to the nearest integer, because $\bar{K}\bar{\phi}^n$ is less than 0.5 and at any rate is exponentially small. By Theorem 8(d), the sequence of shortest LH path lengths $L(2n, 3n)$ for $n = d/2 = 1, 2, 3, \ldots$ is $4, 10, 16, 42, 68, 178, \ldots$, which is the sequence of Fibonacci numbers (multiplied by 2) with every third number omitted. These shortest lengths grow with the square root of the longest lengths, which is still exponential.

COROLLARY 9: *There are $d \times d$ games for even d, where the length of each LH path is at least proportional to $\phi^{3d/4}$.*

A construction using labeling similar to (5) is possible for odd d, but there the path lengths are less symmetric than those in Theorem 8 for even d. We do not need this because it is trivial to obtain an odd-dimensional game from the next lower even dimension by adding a strictly dominated strategy for each player.

4. NONSQUARE GAMES AND SUPPORT ENUMERATION

So far, we have analyzed the $d \times d$ games $\Gamma(d, d)$. They have a unique equilibrium that is found by the LH algorithm after an exponential number of steps for any missing label. However, the equilibrium is completely mixed and is easily found by *support enumeration* (e.g., Dickhaut and Kaplan (1991), Porter, Nudelman, and Shoham (2004), or Bárány, Vempala, and Vetta (2005)). This simple algorithm tests the possible supports of equal size for both players and checks whether equating the expected payoffs to the other player in his support defines mixed strategies that are best responses to each other. There is only one pair of supports where both players use d strategies, so this is tested quickly.

In this section, we consider the $d \times 2d$ games $\Gamma(d, 2d)$. By Lemma 3, in any Nash equilibrium of such a game, both players use mixed strategies with support size d. The following lemma states that for player 2, the supports of equilibrium strategies form only an exponentially small fraction of the possible $\binom{2d}{d}$ supports of size d. The notation $S(d/2)$ is chosen to be consistent with von Stengel (1999).

LEMMA 10: *Let $S(d/2)$ be the set of bit strings of length $2d$ that contain d 1's of the form $s_1 s_2 \cdots s_k$, where each substring s_i is either* 00, 11, *or* 0110. *Then (u, v) is a Nash equilibrium of the game $\Gamma(d, 2d)$ if and only if $u = 0^d s$ and $v = 1^d t$ for $s, t \in S(d/2)$, where $s = s_1 s_2 \cdots s_k$ and $t = t_1 t_2 \cdots t_k$, and t_i is* 11, 00, *or* 0110 *if and only if s_i is* 00, 11, *or* 0110, *respectively, for $1 \le i \le k$. Asymptotically,*

$$\left| S\left(\frac{d}{2}\right) \right| \approx 0.81 \frac{2.414^d}{\sqrt{d}}, \qquad \binom{2d}{d} \approx 0.56 \frac{4^d}{\sqrt{d}}. \tag{17}$$

PROOF: It is easy to see that the described bit strings define Nash equilibria. The claims follow from von Stengel (1999): As in Proposition 3.2 of that paper, one can see that these are the only equilibria. An exact expression for the size of $S(d/2)$ is (3.6) (von Stengel (1999, p. 564)) and von Stengel (1999, p. 566) gives an asymptotic formula, denoted by $\tilde{\sigma}(d/2)$, with rounded parameters as in (17). The expression for $\binom{2d}{d}$ is based on Stirling's formula. *Q.E.D.*

In any Nash equilibrium of the game $\Gamma(d, 2d)$, the strategy of player 1 has full support. The supports of equilibrium strategies of player 2 define the set $S(d/2)$. By (17), these form an exponentially small fraction

$$F = \frac{|S(d/2)|}{\binom{2d}{d}} \approx 1.44 \times 0.6^d \tag{18}$$

of all supports of size d for player 2 (the support of size d for player 1 is unique). This is the success probability F of a support enumeration algorithm that tests a single random support of size d. We can achieve such "random behavior" of any support enumeration algorithm by randomly permuting the columns of the game with a uniformly chosen random permutation. This "hides" the equilibrium supports from the algorithm, so that it has to test an exponential number of supports on average before finding an equilibrium. The algorithm may even know the set $S(d/2)$, called E in the following lemma.

LEMMA 11: *Consider a $d \times 2d$ game where a pair of supports defines a Nash equilibrium if and only both supports have size d, and where player 2's support belongs to the set E, a set of d-sized subsets of $\{1, \dots, 2d\}$. Randomly permute the $2d$ pure strategies of player 2. Then a support enumeration algorithm, even if it knows E, has to test an expected number of*

$$\frac{\binom{2d}{d} - |E|}{|E| + 1} + 1 \tag{19}$$

supports before finding an equilibrium support.

PROOF: By assumption, there is a random permutation p of $\{1, \dots, 2d\}$ so that a d-sized subset T of $\{1, \dots, 2d\}$ is an equilibrium support (for player 2) of the permuted game if and only if $p(T) \in E$. Suppose that the support enumeration algorithm tests such a set T and that T is *not* an equilibrium support. We show that this information does not eliminate any other candidate support: For any $S \in E$, there are exactly $(d!)^2$ permutations q so that $q(T) = S$ and for distinct choices of S these permutations are distinct. Thus, if T is not an equilibrium support, then exactly $|E| \cdot (d!)^2$ permutations q can be excluded from the possible choices for p. Divided by the total number $(2d)!$ of permutations, this is $|E|/\binom{2d}{d}$, which is exactly the "trivial success rate," called F in (18) when $E = S(d/2)$, as claimed.

Consequently, any order of testing d-sized supports is equally good on average. A standard argument (Motwani and Raghavan (1995, p. 10)) then shows that the expected number of support guesses until an equilibrium is found is given by (19), as claimed. *Q.E.D.*

For $E = S(d/2)$, the number in (19) is about 0.7×1.66^d. Support enumeration therefore takes exponential time on average for a game $\Gamma(d, 2d)$ with randomly permuted columns. For the LH algorithm, the permutation does not affect the possible path lengths. The following theorem shows that these are exponentially long, since they are closely related to the paths of the square game $\Gamma(d, d)$, which have length $L(d, k)$ when dropping label k.

THEOREM 12: *Let $M(d, k)$ be the length of the LH path in the game $\Gamma(d, 2d)$ when dropping label k, for $1 \leq k \leq 3d$. Then*
(a) *$M(d, k) = M(d, d + 1 - k)$ and $M(d, d + k) = M(d, 3d + 1 - k)$ for $1 \leq k \leq d$;*
(b) *$M(d, k) = L(d, k)$ for even k and $2 \leq k \leq d$;*
(c) *$M(d, d + k) = L(d, d + k)$ for even k and $1 \leq k \leq d$;*
(d) *$M(d, 2d + k) = L(d, 1) + 1$ for even k and $1 \leq k \leq d$.*

By condition (a) of this theorem, it suffices to consider only even labels k in conditions (b), (c), and (d), so these cover all possible missing labels k. The proof is given in the Appendix.

COROLLARY 13: *There are $d \times 2d$ games, for even d, where the length of each LH path is at least proportional to $\phi^{3d/4} \approx 1.43^d$ and where a support enumeration algorithm has to test on average about 0.7×1.66^d many supports of size d before it finds an equilibrium.*

The games $\Gamma(d, 2d)$ with randomly permuted columns seem to be hard to solve for any known general-purpose algorithm that finds a Nash equilibrium of a bimatrix game. Enumerating the vertices of the polytopes P or Q in (4) is another way to find an equilibrium (see von Stengel (2002) for a survey). A natural starting point for vertex enumeration is the vertex pair $(\mathbf{0}, \mathbf{0})$ of $P \times Q$. In both polytopes, any Nash equilibrium vertex is at least d edges away from that starting point by Lemma 3 and there are an exponential number of such vertices. Therefore, one should expect that finding an equilibrium by vertex enumeration takes long as well. Because there are many vertex enumeration methods, an analysis is clearly beyond the scope of this paper. So far, the games $\Gamma(d, 2d)$ seem hard for any *general* algorithm that finds an equilibrium of a bimatrix game.[3]

[3] Knowing the precise construction of $\Gamma(d, 2d)$, with the complete structure of the polytopes, except for the random permutation of the columns, it is possible to find an equilibrium in $2d + 1$ very special pivoting steps; see Savani (2004). However, these steps are not in any way suitable for finding an equilibrium of a general bimatrix game.

5. SYMMETRIC GAMES AND IMITATION GAMES

Morris (1994) considered "Lemke paths" on simple d-dimensional polytopes T with $2d$ facets. A vertex v of T is given and the d facets incident to v have labels $1, \ldots, d$. The remaining d facets also have labels $1, \ldots, d$, so each label appears twice. A Lemke path starts at v by dropping a label k and traversing the unique edge that leaves the facet with label k. The endpoint of that edge is a new facet that has either label k, which terminates the path, or a duplicate label, which is then dropped by leaving the other facet with that label. The path continues in that manner until another completely labeled vertex is found. Applied to the polytope S as in (1) with vertex $v = \mathbf{0}$, with facet labels $1, \ldots, d$ for the inequalities $z \geq \mathbf{0}$, and $1, \ldots, d$ for $Cz \leq \mathbf{1}$, we have called this the symmetric LH algorithm. Any polytope T with vertex v can be affinely mapped to S with v mapped to $\mathbf{0}$. By Lemma 2, the completely labeled vertices (apart from $\mathbf{0}$) then correspond to the symmetric equilibria of the symmetric bimatrix game $(C, C^\top)$. However, this interpretation was not considered in Morris (1994).

Morris constructed exponentially long Lemke paths by taking for T the dual cyclic polytope in dimension d with $2d$ facets for both odd and even d, suitably labeled as follows. In Section 3, we have identified such a polytope with the set $G(d, 2d)$ of Gale evenness bit strings, which describe the vertices of T in terms of the facets on which they lie. In the order of the bits in those bit strings, the facets $1, \ldots, 2d$ are given the labels

$$(20) \qquad l(k) = k, \quad 1 \leq k \leq d,$$
$$l(d+k) = \begin{cases} d, & k = 1, \\ d-k, & k \text{ even and } 2 \leq k < d, \\ d+2-k, & k \text{ odd and } 2 \leq k \leq d, \\ 1, & k \text{ even and } k = d. \end{cases}$$

For $d = 6$, for example, the 12 facets, corresponding to the positions in a bit string in $G(6, 12)$, are labeled with $1, 2, 3, 4, 5, 6, 6, 4, 5, 2, 3, 1$. Denote the bimatrix games $(C, C^\top)$ obtained from Morris's examples by $\Gamma_M(d)$. Analogous to Corollary 4, it is easy to see that then the only completely labeled vertices of T are 1^d0^d (which is the starting point v of a Lemke path, our artificial equilibrium) and 0^d1^d (which corresponds to a completely mixed symmetric equilibrium).

However, these bimatrix games $\Gamma_M(d)$ have a large number of nonsymmetric equilibria, in particular, two pure strategy equilibria that are always found by the ordinary, nonsymmetric LH algorithm after two or three steps, for *any* missing label. For illustration, we consider the game $\Gamma_M(6)$. The bimatrix game (A, B) is $(C, C^\top)$, so the two polytopes P and Q in (4) are

$$(21) \qquad P = \{x \in \mathbb{R}^d \mid x \geq \mathbf{0},\ Cx \leq \mathbf{1}\}, \quad Q = \{y \in \mathbb{R}^d \mid Cy \leq \mathbf{1}, y \geq \mathbf{0}\},$$

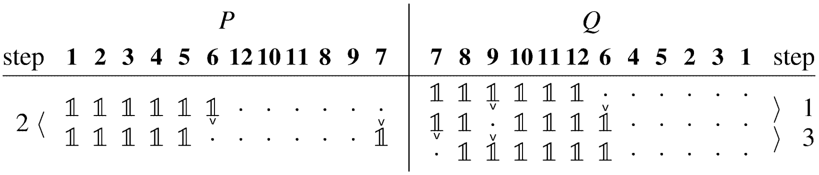

FIGURE 4.—A LH path for the symmetric bimatrix game $\Gamma_M(6)$.

so they are the same polytopes as S in (1), except that the first d inequalities $x \geq \mathbf{0}$ in P are labeled with $1, 2, \ldots, d$, whereas the inequalities $y \geq \mathbf{0}$ in Q are labeled $d+1, d+2, \ldots, 2d$. The inequalities in $Cx \leq \mathbf{1}$ in P correspond to the pure strategies of player 2. Since P is the dual cyclic polytope, this means that the 12 facets of P, as positions in a bit string in $G(6, 12)$, have labels $1, 2, 3, 4, 5, 6, 12, 10, 11, 8, 9, 7$, corresponding to the interpretation of the second d labels in (20) as strategies of player 2. In other words, the labels $12, 10, 11, 8, 9, 7$ mean that the second set of d inequalities that define the dual cyclic polytope appear, respectively, as rows $6, 4, 5, 2, 3, 1$ of C, just as in the symmetric game. In the same way, Q is the dual cyclic polytope, with its 12 facets labeled $7, 8, 9, 10, 11, 12, 6, 4, 5, 2, 3, 1$. Figure 4 shows the LH path for this game with missing label 9. The path terminates at the pure strategy equilibrium $(d, d+1)$, where player 1 plays his last strategy (which has label d) and player 2 plays her first strategy (which has label $d+1$). It is easily shown that in the game $\Gamma_M(d)$, every LH path terminates either at this equilibrium or its symmetric counterpart $(1, 2d)$ if d is even. If d is odd, every LH path of $\Gamma_M(d)$ leads to either $(2, 2d)$ or $(d, d+2)$. For any d, every LH path is only two or three steps long.

The symmetric equilibria of a symmetric game $(C, C^\top)$ correspond to the *arbitrary* equilibria of another, closely related game. This is the *imitation game* (C, I) (introduced in McLennan and Tourky (2005)), which is a square game where the payoff matrix to player 2 is the identity matrix I.

PROPOSITION 14 (McLennan and Tourky (2005)): *The mixed strategy pair* (y, y) *is a Nash equilibrium of the symmetric game* $(C, C^\top)$ *if and only if there is some strategy* x *so that* (x, y) *is a Nash equilibrium of the imitation game* (C, I).

After our construction of the games $\Gamma(d, d)$ described in Savani and von Stengel (2004), McLennan and Tourky (2005) made the ingenious observation that the LH paths for imitation games (C, I), projected to the polytope Q of player 2, give the paths of the symmetric LH algorithm.

PROPOSITION 15 (McLennan and Tourky (2005)): *Let* $(C, C^\top)$ *be a nondegenerate* $d \times d$ *symmetric game* $(C, C^\top)$. *The steps of the symmetric LH algorithm applied to this game with missing label* k, *for* $1 \leq k \leq d$, *correspond exactly to the* *even-numbered steps of the LH path for the imitation game* (C, I) *with missing label k and to the odd-numbered steps of the LH path for* (C, I) *with missing label* $d + k$.

PROOF: For the imitation game (C, I), the polytope Q in (4) is equal to the polytope S in (1). However, the d inequalities $y \geq \mathbf{0}$ in Q have labels $d + 1, \ldots, 2d$ rather than $1, \ldots, d$ in S. The polytope P is the d-cube $\{x \in \mathbb{R}^d \mid x \geq \mathbf{0},\ x \leq \mathbf{1}\}$. Hence, any edge of P drops some label i and picks up label $d + i$, or vice versa, for some $i \in \{1, \ldots, d\}$. Any step of the symmetric LH algorithm is an edge of S and thereby represents an edge of Q. It is easy to see that it corresponds to an even- or odd-numbered step of the LH path for (C, I), as claimed. *Q.E.D.*

Consequently, also noted by McLennan and Tourky (2005), the games $(C, C^\top) = \Gamma_M(d)$ give rise to exponentially long LH paths for the imitation games (C, I). It follows from the results by Morris (1994) that the longest such path has length proportional to $(1 + \sqrt{2})^{d/2} \approx 1.55^d$ and the shortest path has length proportional to $(1 + \sqrt{2})^{d/4} \approx 1.25^d$. For our square games $\Gamma(d, d)$, these numbers are $\phi^{3d/2} \approx 2.06^d$ and, by Corollary 9, $\phi^{3d/4} \approx 1.43^d$, respectively. (For imitation games, the polytope P is the d-cube, so no LH path can have more than 2^d steps.) Thus, the games $\Gamma(d, d)$ have longer LH paths. More significantly, however, they can be extended to the nonsquare games $\Gamma(d, 2d)$ described in the previous section, which are also hard to solve by support enumeration. In contrast, the imitation games (C, I) for $\Gamma_M(d)$, which are necessarily square, are easy to solve by support enumeration.

6. CONCLUSIONS AND OPEN QUESTIONS

The LH algorithm finds one solution to an LCP (2) derived from a bimatrix game. The closely related algorithm by Lemke (1965) solves more general LCPs. Murty (1978) and Fathi (1979) give LCPs where Lemke's algorithm takes exponentially many steps, but these are not derived from games.

A linear program (LP) can be formulated as an LCP, which captures the complementary slackness conditions that characterize a pair of optimal solutions to the primal and dual LP. Applied to such an LCP, Lemke's algorithm corresponds to the self-dual parametric simplex algorithm for solving LPs (Dantzig (1963, p. 245)). A special case of this is a parametric simplex algorithm where the right-hand side is parameterized, which Murty (1980) has shown to be exponential. Another special case is the parametric-objective simplex algorithm, which Goldfarb (1983, 1994) has shown to be exponential.

Equilibria of zero-sum games are the solutions to an LP. The results by Murty (1980) and Goldfarb (1983, 1994) suggest that Lemke's algorithm can be exponential even when solving a zero-sum game. However, these results do not extend to the LH algorithm. First, the examples by Murty and Goldfarb define a single path of Lemke's algorithm that is not an LH path. Second, even if they could be modified to define an LH path, the LP defines a cube as in the construction by Klee and Minty (1972). The endpoints of the exponentially long path on the cube are joined by a single edge. Hence, even if one could make the LH algorithm mimic that path, another LH path would be very short.

The LCP description of an LP is a useful tool for analyzing the *expected* running time of the simplex algorithm. For various models of random input data, that expected running time is polynomial, in contrast to the worst-case exponential behavior (see Todd (2001, p. 422) for a survey). Smale (1983) and Adler and Megiddo (1985) give a probabilistic analysis for the self-dual parametric simplex algorithm, using its description as a special case of Lemke's algorithm.

This raises the following questions in the context of games. First, what is the expected running time of the LH algorithm? Megiddo (1986) analyzed Lemke's algorithm for random general LCPs (not derived from games), and showed that its expected running time is exponential for the standard version of Lemke's algorithm and quadratic for a modified version. Bárány, Vempala, and Vetta (2005) showed that with high probability, random games have equilibria with small support, so that an equilibrium is quickly found by support enumeration. Although this result does not concern the LH algorithm, it suggests that random games are not hard to solve.

Second, is there a randomized variant of Lemke's algorithm that solves our games quickly on average? This possibility is suggested by a certain parameter, called the *covering vector*, that can be freely chosen in Lemke's algorithm. Megiddo's (1986) quadratic expected running time for random LCPs applies to Lemke's algorithm with a special covering vector. Von Stengel, van den Elzen, and Talman (2002) used a covering vector derived from a starting pair of mixed strategies to solve two-player games with Lemke's algorithm and gave a game-theoretic interpretation of the resulting path. The starting pair can be chosen randomly. More general random choices of the covering vector are also possible. This method is currently being studied using a new unified view of LH and Lemke's algorithm (see Savani (2005)).

A different open problem is how to generate "numerically stable" game matrices with our construction. We construct cyclic polytopes with the moment curve, which gives rise to notoriously ill-conditioned matrices. As a consequence, numerical problems arise when the pivoting steps are implemented using floating-point arithmetic. It would be good to have "hard instances" of games without this additional complication. These numerical problems may possibly be avoided by using points on the so-called trigonometric moment curve (see Ziegler (1995, p. 75) or Grünbaum (2003, p. 67)). An open question is the required numerical accuracy of these points.

We conclude with a 6 × 6 game (A, B) that is an example of $\Gamma(6,6)$, using dual cyclic polytopes derived from the trigonometric moment curve:

$$A = \begin{bmatrix} -180 & 72 & -333 & 297 & -153 & 270 \\ -30 & 17 & -33 & 42 & -3 & 20 \\ -81 & 36 & -126 & 126 & -36 & 90 \\ 90 & -36 & 126 & -126 & 36 & -81 \\ 20 & -3 & 42 & -33 & 17 & -30 \\ 270 & -153 & 297 & -333 & 72 & -180 \end{bmatrix},$$

$$B = \begin{bmatrix} 72 & 36 & 17 & -3 & -36 & -153 \\ -180 & -81 & -30 & 20 & 90 & 270 \\ 297 & 126 & 42 & -33 & -126 & -333 \\ -333 & -126 & -33 & 42 & 126 & 297 \\ 270 & 90 & 20 & -30 & -81 & -180 \\ -153 & -36 & -3 & 17 & 36 & 72 \end{bmatrix}.$$

Here, the LH algorithm finds the unique completely mixed equilibrium in 88 steps when dropping label 1, which gives the longest path, and in 16 steps when dropping label 9, which gives the shortest path. Using the same matrix B, the bimatrix game (A', B) with

$$A' = \begin{bmatrix} -81 & 36 & -126 & 126 & -36 & 90 \\ -180 & 72 & -333 & 297 & -153 & 270 \\ 20 & -3 & 42 & -33 & 17 & -30 \\ -30 & 17 & -33 & 42 & -3 & 20 \\ 270 & -153 & 297 & -333 & 72 & -180 \\ 90 & -36 & 126 & -126 & 36 & -81 \end{bmatrix}$$

has 75 equilibria. It is the smallest known example of a nondegenerate $d \times d$ game with more than $2^d - 1$ equilibria, refuting a conjecture by Quint and Shubik (1997). As shown by von Stengel (1999), such games can be constructed with dual cyclic polytopes by using a different labeling of the second polytope that is similar to the permutation l' in (5). Note that A' is obtained from A by permuting rows. The construction of A, B, and A' is explained in detail in Appendix A of Savani and von Stengel (2004).

Dept. of Mathematics, London School of Economics, London WC2A 2AE, United Kingdom; rahul@maths.lse.ac.uk

and

Dept. of Mathematics, London School of Economics, London WC2A 2AE, United Kingdom; stengel@maths.lse.ac.uk.

Manuscript received May, 2004; final revision received July, 2005.

APPENDIX

In this appendix, we prove Lemma 6 and Theorems 7, 8, and 12. We also state several auxiliary definitions and lemmas used in those proofs.

Suppose that the bit strings $x = x_1x_2\cdots x_{2d}$ and $y = y_1y_2\cdots y_{2d}$ are two vertices of $G(d)$. Then they are connected by an edge if and only if they differ only by a substring of the form 1^p0 in x and 01^p in y or vice versa, for some even $p \geq 2$. That is, there are two positions i and j so that

$$\{x_ix_{i+1}\cdots x_j,\ y_iy_{i+1}\cdots y_j\} = \{1^p0,\ 01^p\}.$$

(If $j < i$, this uses the cyclic symmetry of the Gale evenness bit strings, taking $2d + 1$ as 1 as in step 1 in Q in Figure 2.) We say that the edge *crosses* the positions $i+1, \ldots, j-1$. For example, the vertices x and y of P joined by step 3 in Figure 1 are 011110000 and 01101100, where $i = 4$, $j = 6$, $x_4x_5x_6 = 110$, $y_4y_5y_6 = 011$, and the remaining positions of x and y are the same. This edge crosses only position 5. Recall that edges of LH paths are edges of the product polytope $P \times Q$, joining (u, v) to (u', v) or (u, v) to (u, v'). We say that such an edge crosses a position k if this holds for the respective edge joining u to u' in P or v to v' in Q.

To emphasize the dimension d, we write

$$e_0^d = (1^d0^d, 0^d1^d), \quad e_1^d = (0^d1^d, 1^d0^d).$$

LEMMA 16: *No edge of $\pi(d, 1)$ crosses position* 1 *or* $2d$.

PROOF: The first edge of $\pi(d, 1)$ joins e_0^d to $(01^d0^{d-1}, 0^d1^d)$ and does not cross position 1 or $2d$. The same holds for the last edge that joins $(0^d1^d, 01^d0^{d-1})$ to e_1^d. In any other edge, the bit in position 1 is zero in both polytopes, so the edge cannot cross position 1 or the cyclically adjacent position $2d$. *Q.E.D.*

The next lemma concerns the first and last vertex pair of the subpath $B(d)$ of $\pi(d, 2d)$ defined in Lemma 6.

LEMMA 17: *Let $L = L(d, 2d)$ and let (u^i, v^i) be the ith vertex pair of the path $\pi(d, 2d)$ for $0 \leq i \leq L = L(d, 2d)$. Then*

$$(u^1, v^1) = (1^d0^d, 10^{d-1}1^{d-2}01), \tag{22}$$

$$(u^{L-2}, v^{L-2}) = (0^{d-1}1^d0, 1^{d-1}0^d1). \tag{23}$$

PROOF: See Figure 2 for an illustration of the case $d = 6$. After step 1 of $\pi(d, 2d)$, the vertex pair (u^1, v^1) as in (22) is reached. The last vertex pair of $\pi(d, 2d)$ is $(u^L, v^L) = e_1^d$. This is reached in step $L - 1$ by picking up label $2d$

in P. Hence, the previous vertex pair is $(u^{L-1}, v^{L-1}) = (0^{d-1}1^d0, 1^d0^d)$, where label d is duplicate. The vertex pair (u^{L-2}, v^{L-2}) is therefore as in (23). *Q.E.D.*

The next lemma states conditions similar to (7) and (9) in Lemma 6, except that they concern labels rather than positions of bits. For the polytope Q, these conditions therefore do not describe the point-symmetry visible in Figure 2.

LEMMA 18: *Let $L = L(d, 2d)$ and let (u^i, v^i) be the ith vertex pair of the path $\pi(d, 2d)$ for $0 \le i \le L$. Let r be the permutation of $\{1, \dots, 2d\}$ defined by $r(k) = 2d - k$ for $1 \le k \le 2d - 1$ and $r(2d) = 2d$. Then for $2 \le i \le L - 2$, step i of $\pi(d, 2d)$ corresponds to step $L - i$ as follows: If label k is duplicate in the vertex pair (u^{i-1}, v^{i-1}) and hence is dropped in step i to get (u^i, v^i), then label $r(k)$ is duplicate in (u^{L-i}, v^{L-i}) and hence is picked up when reaching that vertex pair from (u^{L-1-i}, v^{L-1-i}) in step $L - i$. Furthermore, for $1 \le i \le L - 2$,*

$$u^i_k = u^{L-1-i}_{r(k)}, \quad 1 \le k \le 2d, \tag{24}$$

$$v^i_{l'(k)} = v^{L-1-i}_{l'(r(k))}, \quad 1 \le k \le 2d. \tag{25}$$

PROOF: The proof will be by induction on i. The permutation r serves as a relabeling to state (and prove) the symmetry of the path. It writes the labels $1, \dots, 2d - 1$ backwards and keeps the missing label $2d$ fixed. For a vertex u of P, labels and positions are the same. Then r, seen as a reversal of the bit strings and cyclic shift by one position, preserves the Gale evenness condition. The induction is therefore easy for P. Equations (22) and (23) clearly imply (24) for $i = 1$.

For a vertex v of Q, equation (25) has to be read as follows: The bit of vertex v^i in position $l'(k)$, which has label k (because l' is its own inverse), is equal to the bit of vertex v^{L-1-i} in position $l'(r(k))$, which has label $r(k)$.

The set $\{2, 3, \dots, 2d - 2\}$ is mapped to itself under both r and l'. Both bijections map d to itself and it is easy to see that

$$l'(r(k)) = r(l'(k)), \quad 2 \le k \le 2d - 2. \tag{26}$$

Consequently, for the more restrictive case $2 \le k \le 2d - 2$, equation (25) is equivalent to

$$v^i_{l'(k)} = v^{L-1-i}_{r(l'(k))}, \quad 2 \le k \le 2d - 2, 1 \le i \le L - 2. \tag{27}$$

Equation (27) implies that r can be applied to the *positions* of the bit string v that belong to the set $\{l'(2), \dots, l'(2d - 2)\}$, which is equal to $\{2, \dots, 2d - 2\}$. If r could be applied to all positions of v, then r would be directly "compatible" with both the labels and the adjacency of vertices in Q. However, for $k \in \{1, 2d - 1, 2d\}$, equation (26) does not hold, which complicates our proof.

We have

$$\begin{array}{lll} l'(1)=1, & l'(r(1))=2d, & r(l'(1))=2d-1, \\ l'(2d-1)=2d, & l'(r(2d-1))=1, & r(l'(2d-1))=2d, \\ l'(2d)=2d-1, & l'(r(2d))=2d-1, & r(l'(2d))=1. \end{array} \tag{28}$$

As Figure 2 shows, the positions $1, 2d-1$, and $2d-2$ in Q are constant. Indeed, we will show

$$v_1^i = 1, \quad v_{2d-1}^i = 0, \quad v_{2d}^i = 1, \quad 1 \le i \le L-2. \tag{29}$$

The left two columns of (28) then show (25) also for $k = 1, 2d-1, 2d$. Instead of (25), we prove by induction on i the stronger assertions (29) and (27). By (22) and (23), they are true for $i = 1$.

The length L of $\pi(d, 2d)$ is even because the path starts in Q and ends in P. Hence, if i is odd or even, so is $L-i$.

As inductive hypothesis, suppose that (24), (29), (27), and therefore (25), hold for $i-1$ instead of i. By the above considerations, this implies that if k is the duplicate label of (u^{i-1}, v^{i-1}), to be dropped in step i, then $r(k)$ is the duplicate label of (u^{L-i}, v^{L-i}), to be picked up in step $L-i$.

Suppose first that i is even. Then step i from (u^{i-1}, v^{i-1}) to (u^i, v^i) is in P, that is, the duplicate label is dropped in u^{i-1} to give the new vertex u^i, and $v^{i-1} = v^i$. Thus, (29) and (27) hold trivially for i. Because r preserves Gale evenness, the edge that connects u^{i-1} to u^i in P corresponds to the edge that connects u^{L-i} to u^{L-1-i} as described. Hence, (24) holds for i, which completes the induction step for even i.

Second, let i be odd, where step i is in Q. Let k be the duplicate label of (u^{i-1}, v^{i-1}). In step i, label k is dropped in v^{i-1} to give the new vertex v^i, and $u^{i-1} = u^i$. Hence, (24) holds trivially for i. If $k = 1$, then because (29) holds for $i-1$ by inductive hypothesis, changing $v_{l'(k)}^{i-1}$ to zero would give $v_{l'(2d)}^i = 1$ and thereby terminate the path, which is not possible. Hence, $k \neq 1$. Similarly, if $k = 2d-1$, then since (25) holds by inductive hypothesis, label $r(k)$ is duplicate in (u^{L-i}, v^{L-i}), where $r(k) = 1$ and so label $2d$ would be picked up in (u^{L-1-i}, v^{L-1-i}) when going along the edge in Q from v^{L-i} to v^{L-1-i}. So $k \neq 2d-1$, which shows the induction step for (29).

The duplicate label k therefore fulfills $2 \le k \le 2d-2$ (since obviously $k \neq 2d$) and so does the label that is picked up, where the positions 1, $2d-1$, and $2d$ are not crossed because of (29). By inductive hypothesis, (27) holds for $i-1$, and r preserves the adjacency of positions. Hence, step i in Q corresponds to the backward step $L-i$ in Q as claimed, which shows that equation (27) also holds for i. This completes the induction. *Q.E.D.*

PROOF OF LEMMA 6: The preceding proof also shows Lemma 6. Lemma 18 specifies the statement to be proved by induction and concerns the labels. We also used the fact that the permutation r preserves the adjacency of positions, as stated in the equations of Lemma 6. We have shown these because equation (7) is equivalent to (24), equation (8) follows from (29), equation (9) is equivalent to (27), and equation (10) holds trivially. Label 1 is never again picked up in P because it is present in Q from step 1 onward, by (29). *Q.E.D.*

PROOF OF THEOREM 7: Overview: The mappings are given as follows: β and β' are defined on $G(d) \times G(d)$, where

$$\beta(u, v) = (u, 0v_2v_3 \cdots v_{2d-2}1v_{2d}). \tag{30}$$

The mapping β' applies to a final segment of the path $A(d)$, which is symmetric as stated in Lemma 5. Hence, β' is determined by β, which applies to an initial segment of $A(d)$ (see (33)). The other mappings are α, γ, $\gamma' : G(d-2) \times G(d-2) \to G(d) \times G(d)$. With $\overleftarrow{u}$ defined as the bit string u reversed,

$$\alpha(u, v) = (0\overleftarrow{u}110, 0\overleftarrow{v}110). \tag{31}$$

With $c = 2d - 4$,

$$\gamma(u_1 \cdots u_c, v) = (u_1 11 u_2 \cdots u_c 00, 10v01). \tag{32}$$

The mapping γ' in (13) applies to a final segment of $B(d)$, which, as we will show, starts after the midpoint of $B(d)$. Thus, γ' is determined by γ due to the symmetry of $B(d)$ stated in Lemma 6.

First we show the following equation, which is equivalent to (11) and (12):

$$A(d) = \beta(B(d)) + \alpha(A(d-2)) + \beta'(B(d)). \tag{33}$$

Note that only positions 1 and $2d - 1$ in Q (corresponding to the missing label in $A(d)$ and $B(d)$, respectively) are changed by the mapping β, and these positions are constant throughout $B(d)$ by (8) and (10). The starting point (u^1, v^1) of $B(d)$ is given by (22), and, in the first step of $B(d)$, label 1 is dropped in P. The path $A(d)$ is also started by dropping label 1 in P from e_0^d. Now $\beta(u^1, v^1) = e_0^d$ as required and, in the first step of $A(d)$ and $B(d)$, the label to be dropped is 1 in P. Because (u^1, v^1) and e_0^d differ only in positions that are constant throughout $B(d)$, the path $B(d)$ maps to $\beta(B(d))$ and thereby represents the initial part of $A(d)$. For $d = 6$, Figures 2 and 3 show how steps 2–34 of $\pi(6, 12)$ map to steps 1–33 of $A(6)$ in this way. By (23), the endpoint of $\beta(B(d))$ is

$$\beta(0^{d-1}1^d0, 1^{d-1}0^d1) = (0^{d-1}1^d0, 01^{d-2}0^{d-1}11). \tag{34}$$

The duplicate label is $2d - 1$, which has been picked up in P. So in the next step of $A(d)$ (which is step 34 in Figure 3), label $2d - 1$ is dropped in Q and label $2d - 3$ is picked up, giving the vertex pair

$$(u^*, v^*) = (0^{d-1}1^d0, 01^{d-2}0^{d-2}110). \tag{35}$$

(For the path $\pi(d, 2d)$, label d would be picked up instead at this stage, as in step 35 in Figure 2.) This is the edge of $A(d)$ that joins $\beta(B(d))$ to $\alpha(A(d-2))$ in (33).

We are now at the start of $C(d)$ and want to show that this path segment starts with $\alpha(A(d-2))$ with α in (31). Indeed, the starting vertex pair of $C(d)$ is $(u^*, v^*) = \alpha(e_0^{d-2})$. The duplicate label is $2d-3$, which is to be dropped in P in the next step (step 35 in Figure 3). The subsequent steps are represented by $\alpha(A(d-2))$, because in the lower-dimensional polytope, label 1, which is mapped by α to label $2d-3$ of the higher-dimensional polytope, is dropped; here, we consider α also as an injective map of labels, obtained in the obvious way from (31), namely $\alpha(k) = 2d-2-k$ for $1 \le k \le 2d-4$. Essentially, the subsequent steps in $A(d-2)$ map into higher dimension by (31) and by Lemma 16; we only need to check complementarity of the constant positions in higher dimension. In the higher dimension, position 1 with the missing label 1 is zero in both polytopes, consistent with (31). Positions $2d-1$ and $2d$ are also complementary by (31). For positions $2d-3$ and $2d-2$, we have complementarity because $2d-3$ is zero, because it is obtained from the position with the missing label 1 in lower dimension. This shows that the initial segment of $C(d)$ is indeed $\alpha(A(d-2))$.

In the last step of $A(d-2)$, label 1 is picked up in Q (this is step 20 in Figure 1), so in the last step of $\alpha(A(d-2))$, label $2d-2$ is picked up in Q (this is step 54 in Figure 3). Then we are at the vertex pair $(v^*, u^*) = \alpha(e_1^{d-2})$, which is $(01^{d-2}0^{d-2}110, 0^{d-1}1^d0)$ by (35). We have shown that the initial part of $A(d)$ in (33) is $\beta(B(d)) + \alpha(A(d-2))$ and that the starting point and endpoint of $\alpha(A(d-2))$ are (u^*, v^*) and (v^*, u^*), respectively. Then the rest of the path $A(d)$ in (33) is obtained by Lemma 5: The next vertex pair, obtained from (v^*, u^*) by dropping label $2d-2$ in P (step 55 in Figure 3), is

$$(36) \qquad (u', v') = (01^{d-2}0^{d-1}11, 0^{d-1}1^d0),$$

which agrees with Lemma 5 and its symmetric counterpart in (34) (in Figure 3, step 34 backwards). Thus, the remainder is the path $\beta(B(d))$ backwards but with the bit strings for P and Q exchanged. Using the symmetry of $B(d)$ in Lemma 6, this part of the path can be expressed as $\beta'(B(d))$ with a suitably defined mapping β', similar to β, which exchanges the bit strings for P and Q. This shows (33).

We now show (13). The first part of $B(d)$ is indeed $\gamma(A(d-2))$: Both $B(d)$ and $A(d-2)$ start by dropping label 1 in P, and the starting point of $B(d)$ is $\gamma(e_0^{d-2})$. Then $B(d)$ proceeds like $\gamma(A(d-2))$ because of Lemma 16 and since complementarity holds for the constant positions in higher dimension, which is easily checked using (32). Next, by (33),

$$(37) \qquad \gamma(A(d-2)) = \gamma[\beta(B(d-2)) + \alpha(A(d-4)) + \beta'(B(d-2))].$$

Now consider the starting point (u'', v'') of $\beta'(B(d-2))$, which is (u', v') given by (36) but with $d-2$ instead of d. (For $d = 6$, (u'', v'') is the start of step 14 of $A(4)$ in Figure 1.) Furthermore, consider the endpoint of $\beta'(B(d-2))$, that is, the endpoint e_1^{d-2} of $A(d-2)$. The images of these points under γ are

$$\begin{aligned}\gamma(u'',v'') &= \gamma(01^{d-4}0^{d-3}11, 0^{d-3}1^{d-2}0)\\ &= (01^{d-2}0^{d-3}1100, 10^{d-2}1^{d-2}001),\\ \gamma(e_1^{d-2}) &= \gamma(0^{d-2}1^{d-2}, 1^{d-2}0^{d-2}) = (0110^{d-3}1^{d-2}00, 101^{d-2}0^{d-1}1).\end{aligned}$$

(For $d=6$, these two points are, respectively, the beginning of step 15 and the end of step 21 of $\pi(6,12)$ in Figure 2, corresponding to steps 14 and 20 of $B(6)$.) These two vertex pairs $\gamma(u'',v'')$ and $\gamma(e_1^{d-2})$ are point-symmetric images of each other under the symmetry of $B(d)$ described in Lemma 6. This means that the endpoint $\gamma(e_1^{d-2})$ of $\gamma(A(d-2))$ is already in the second half of $B(d)$. The central part of $B(d)$ (steps 15–21 in Figure 2), given by the last part of $\gamma(A(d-2))$ in (37), is $\gamma[\beta'(B(d-2))]$. Therefore, there is a mapping γ' so that

$$\begin{aligned}B(d) = \gamma\big[&\beta(B(d-2)) + \alpha(A(d-4)) + \beta'(B(d-2))\big]\\ &+ \gamma'\big[\alpha(A(d-4)) + \beta'(B(d-2))\big],\end{aligned}$$

because the paths $A(d-4)$ and $B(d-2)$ are symmetric and therefore do not have to be written backwards. This representation of $B(d)$ is equivalent to (13), as claimed. *Q.E.D.*

For any paths A and B on $G(d)\times G(d)$, considered as sequences of vertex pairs, let AB denote the path A joined to the path B, where the endpoint of A is equal to the starting point of B. The length (number of edges) of AB is the sum of the lengths of A and B. The next lemma uses this concatenation of paths.

LEMMA 19: *Let k be even and let $2\le k\le d-2$. Then*

(38) $$\pi(d,k) = \alpha(A(k))\beta(A(d-k))$$

with $\alpha: G(k)\times G(k)\to G(d)\times G(d) = P\times Q$,

(39) $$\alpha(u,v) = (u_k\cdots u_1 1^{d-k}0^{d-k}u_{2k}\cdots u_{k+1}, v_k\cdots v_1 0^{d-k}1^{d-k}v_{2k}\cdots v_{k+1}),$$

and $\beta: G(d-k)\times G(d-k)\to G(d)\times G(d)$,

(40) $$\beta(u,v) = (0^k u 1^k, 1^k v 0^k).$$

Furthermore, no edge of the path $\pi(d,k)$ crosses position $2d-k$ or $2d-k+1$.

PROOF: The starting point of $\pi(d,k)$ is e_0^d. As required, $\alpha(e_0^k)=e_0^d$. In the first step of $\pi(d,k)$, label k is dropped in P. Position 1 of the lower-dimensional polytopes, given by the bits u_1 and v_1 in (39), is mapped by α to position k in both P and Q in the higher dimension. In P, position k has label k, which is missing in $\pi(d,k)$. This missing label in the higher-dimensional polytope P corresponds to the missing label 1 in the lower dimension. The last position $2k$ in the lower dimension, with bits u_{2k} and v_{2k}, is mapped to position $2d-k+1$ in the higher dimension. By Lemma 16, no edge of the path $A(k)$ crosses the positions 1 and $2k$ in the lower dimension, so inserting the substrings $1^{d-k}0^{d-k}$ or $0^{d-k}1^{d-k}$ between these bits as done in (39) gives edges in P and Q, respectively. Furthermore, no edge of $\alpha(A(k))$ crosses position $2d-k$ or position $2d-k+1$.

The mapping α preserves the cyclic adjacency of labels. The first $L(k,1)$ steps of $\pi(d,k)$ are given by $\alpha(A(k))$ if the positions $k+1$ up to $2d-k$ of the higher-dimensional polytopes are complementary. Complementarity of positions $k+2,\dots,2d-k$ is immediate. Positions k and $k+1$ in P and Q, respectively, correspond to the missing label k of $\pi(d,k)$ and are thus both zero throughout. Finally, the bit in position $k+1$ in P, with label $k+1$, is 1 according to (39). That bit is complementary to the bit in position k in Q, which has label $k+1$, since this bit corresponds to the missing label in the lower-dimensional polytope and is therefore zero throughout.

After $L(k,1)$ many steps, the vertex pair $\alpha(e_1^k)$ is reached, where

$$\alpha(e_1^k)=(0^k1^{d-k}0^{d-k}1^k,1^k0^{d-k}1^{d-k}0^k)=\beta(e_0^{d-k}).$$

This is also the starting point of $\beta(A(d-k))$, as required. In a similar way as before, one can see that this is the second part of $\pi(d,k)$, which ends in $\beta(e_1^{d-k})=e_1^d$. Lemma 16 for the lower dimensional polytope and equation (40) imply that no edge of $\beta(A(d-k))$ crosses position $2d-k$ or $2d-k+1$. *Q.E.D.*

LEMMA 20: *Let k be even and let $2\le k\le d$. Then*

$$\pi(d,d+k)=e_0^d+\gamma(B(k))\delta(B(d-k+2))+e_1^d, \tag{41}$$

where $\gamma: G(k)\times G(k)\to G(d)\times G(d)=P\times Q$,

$$\gamma(u,v)=(u_1 1^{d-k}u_2\cdots u_{2k}0^{d-k},v_1 0^{d-k}v_2\cdots v_{2k}1^{d-k}), \tag{42}$$

and $\delta: G(d-k+2)\times G(d-k+2)\to G(d)\times G(d)=P\times Q$,

$$\begin{aligned}\delta(u,v)=\big(&v_{d-k+2}\cdots v_{2d-2k+2}0^{k-2}1^k0v_2\cdots v_{d-k+1},\\ &u_{d-k+2}\cdots u_{2d-2k+2}1^{k-2}u_{2d-2k+3}0^{k-1}u_1\cdots u_{d-k+1}\big).\end{aligned}$$

No edge of $\pi(d,d+k)$ crosses position $d+k$ or $d+k+1$ in P.

PROOF: According to (41), $\pi(d, d+k)$ is the concatenation of two paths in dimensions k and $d-k+2$. These do not sum to d, unlike in (38). This works because in the vertex pair $\delta(u, v)$, the vertex in P is obtained from v by ignoring the bits v_1, $v_{2d-2k+3}$, which are constant throughout $B(d-k+2)$ by Lemma 6, and by adding $2k-1$ constant bits, and the vertex in Q is obtained from u by ignoring the bit $u_{2d-2k+4}$ and by adding $2k-3$ constant bits.

It can be verified that both γ and δ preserve the adjacency of the labels used by the LH path and complementarity. The path $\pi(d, d+k)$ starts as follows: In step 1, label $d+k$ is dropped from e_0^d in Q. The new vertex pair is $(1^d0^d, 10^{d-1}1^{k-2}01^{d-k+1})$, which is equal to $\gamma(u^1, v^1)$ for the first vertex pair (u^1, v^1) of $B(k)$ as in (22) (with k instead of n). Then the path continues as described in (42) because, by Lemma 6, it first drops label 1 in P and never picks it up again, and because the bits v_1 and v_{2k} in (42) stay constant according to (8). The last vertex pair of $\gamma(B(k))$ is, by (23) and (42), equal to

$$\gamma(0^{k-1}1^k0, 1^{k-1}0^k1) = (01^{d-k}0^{k-2}1^k0^{d-k+1}, 10^{d-k}1^{k-2}0^k1^{d-k+1}).$$

This is equal to $\delta(1^{d-k+2}0^{d-k+2}, 10^{d-k+1}1^{d-k}01)$, which is δ applied to the first vertex pair of $B(d-k+2)$, using (22) with $d-k+2$ instead of d. The duplicate label is $d+k-1$ and is to be dropped in Q. The corresponding bit is in position $d-k$ in Q and is the image of the bit u_1 under δ. As stated at the end of Lemma 6, this bit u_1 is indeed changed to zero in the first step of $B(d-k-2)$. The last vertex pair of $\delta(B(d-k+2))$ is

$$\delta(0^{d-k+1}1^{d-k+2}0, 1^{d-k+1}0^{d-k+2}1) = (0^{d-1}1^k01^{d-k}, 1^d0^d)$$

with duplicate label d. This label has just been picked up in Q and the corresponding bit in position d is the image of bit $u_{2d-2k+3}$ under δ. When label d is then dropped in P, the endpoint e_1^d is reached, which terminates the path $\pi(d, d+k)$. This completes the proof of (41). *Q.E.D.*

PROOF OF THEOREM 8: For (a), let ψ be defined by $\psi(k) = d-k+1$ and $\psi(d+k) = 2d-k+1$ for $k = 1, \dots, d$. This is a cyclic shift by d followed by a reversal of positions, which leaves the set $G(d)$ invariant and maps e_0^d to itself. Furthermore, ψ commutes with the labelings l and l' of P and Q. Consequently, when the positions of the bit strings that represent the vertex pairs on the path $\pi(d, k)$ are permuted by ψ, one obtains the path $\pi(d, d-k+1)$, which has therefore the same length as $\pi(d, k)$.

To show (b), let $2 \le k \le d-2$. As in Lemma 5, the relabeling l' in (5) applied to both P and Q shows that $\pi(d, k)$ corresponds to the path $\pi(d, k+1)$ backwards, so these paths have the same length.

Claim (c) follows from (38) in Lemma 19.

According to (41) in Lemma 20, the length of $\pi(d, d+k)$ is the sum of the lengths of $B(k)$ and of $B(d-k+2)$ plus 2 (for the first edge from e_0^d and last edge to e_1^d), which shows (d). *Q.E.D.*

Proof of Theorem 12: Claim (a) is proved in the same way as Theorem 8(a), with the permutation ψ of $\{1, \ldots, 3d\}$ defined by $\psi(k) = d + 1 - k$ for $1 \leq k \leq d$ and $\psi(d + k) = 3d + 1 - k$ for $1 \leq k \leq 2d$.

For (b), (c), and (d), let k be even and let $2 \leq k \leq d$. We consider the mappings $\varepsilon, \zeta, \eta : P \times Q = G(d, 2d) \times G(d, 2d) \to G(d, 3d) \times G(2d, 3d)$ defined by

$$\varepsilon(u, v) = (u_1 \cdots u_{2d-k} 0^d u_{2d-k+1} \cdots u_{2d}, v_1 \cdots v_{2d-k} 1^d v_{2d-k+1} \cdots v_{2d}), \tag{43}$$

$$\zeta(u, v) = (u_1 \cdots u_{d+k} 0^d u_{d+k+1} \cdots u_{2d}, v_1 \cdots v_{d+k} 1^d v_{d+k+1} \cdots v_{2d}), \tag{44}$$

$$\eta(u, v) = (u 0^d, 1 v_2 \cdots v_{2d} 1^{k-2} v_1 1^{d-k+1}). \tag{45}$$

Let $\rho(d, j)$ be the LH path for the game $\Gamma(d, 2d)$ with missing label j for any $j = 1, \ldots, 3d$. We show that $\rho(d, k) = \varepsilon(\pi(d, k))$. Both paths start by dropping the same label k. If $k < d$, then by Lemma 19, no edge of the path $\pi(d, k)$ crosses position $2d - k$ or $2k - k + 1$. This holds also when $k = d$, because by Theorem 8(a), $\pi(d, d)$ corresponds to $\pi(d, 1)$, and Lemma 16 implies that $\pi(d, d)$ does not cross positions d and $d + 1$. Therefore, it is possible to insert between positions $2d - k$ and $2k - k + 1$ the bit string 0^d in P and 1^d in Q, as in (43), which implies $\rho(d, k) = \varepsilon(\pi(d, k))$, as claimed. This proves (b).

Next, we show that $\rho(d, d + k) = \zeta(\pi(d, d + k))$. By Lemma 20, no edge of $\pi(d, d + k)$ crosses position $d + k$ or $d + k + 1$ in P, so it is possible to insert the bit string 0^d between these positions, as done in (44). To obtain complementarity, 1^d is inserted between positions $d + k$ and $d + k + 1$ in Q. These positions in Q are crossed by some edges of the path $\pi(d, d + k)$, but because a contiguous string of 1's is inserted, those steps of $\pi(d, d + k)$ are mapped by ζ to the respective steps of $\rho(d, d + k)$ as well. This shows the claim, giving (c).

Finally, we show $\rho(d, 2d + k) = (1^d 0^{2d}, 0^d 1^{2d}) + \eta(\pi(d, 1))$. Dropping label $2d + k$, the first step of $\rho(d, 2d + k)$ reaches the vertex pair $(1^d 0^{2d}, 10^{d-1} 1^d 1^{k-2} 0 1^{d-k+1})$, which is equal to $\eta(e_0^d)$. In this vertex pair, the duplicate label is 1, which is dropped in P in the next step 2 of $\rho(d, 2d + k)$. Beginning with this step, $\rho(d, 2d + k)$ is equal to $\eta(\pi(d, 1))$ (from step 1 onward) for the following reasons: The vertex pairs are almost complementary; by Lemma 16, the bit string 0^d can be inserted at the end of u in (45), and 1^{k-2} can be inserted between v_{2d} and v_1; finally, 1^{d-k+2} can be cyclically inserted between positions v_1 and v_2, which does not affect the steps in the second polytope. This proves (d). *Q.E.D.*

REFERENCES

Adler, I., and N. Megiddo (1985): "A Simplex Algorithm whose Average Number of Steps Is Bounded Between Two Quadratic Functions of the Smaller Dimension," *Journal of the Association for Computing Machinery*, 32, 871–895.

BÁRÁNY, I., S. VEMPALA, AND A. VETTA (2005): “Nash Equilibria in Random Games,” in *Proceedings of the 46th IEEE Symposium on Foundations of Computer Science (FOCS)*. New York: IEEE Press, 123–131.

CODENOTTI, B., AND D. ŠTEFANKOVIČ (2005): “On the Computational Complexity of Nash Equilibria for $(0,1)$ Bimatrix Games,” *Information Processing Letters*, 94, 145–150.

CONITZER, V., AND T. SANDHOLM (2003): “Complexity Results about Nash Equilibria,” in *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)* 765–771.

COTTLE, R. W., J.-S. PANG, AND R. E. STONE (1992): *The Linear Complementarity Problem*. San Diego, CA: Academic Press.

DANTZIG, G. B. (1963): *Linear Programming and Extensions*. Princeton, NJ: Princeton University Press.

DICKHAUT, J., AND T. KAPLAN (1991): “A Program for Finding Nash Equilibria,” *The Mathematica Journal*, 1 (4), 87–93.

FATHI, Y. (1979): “Computational Complexity of LCPs Associated with Positive Definite Symmetric Matrices,” *Mathematical Programming*, 17, 335–344.

GALE, D. (1963): “Neighborly and Cyclic Polytopes,” in *Convexity*, Proceedings of Symposia in Pure Mathematics, Vol. 7, ed. by V. Klee. Providence, RI: American Mathematical Society, 225–232.

GAREY, M. R., AND D. S. JOHNSON (1979): *Computers and Intractability: A Guide to the Theory of NP-Completeness.* San Francisco, CA: Freeman.

GILBOA, I., AND E. ZEMEL (1989): “Nash and Correlated Equilibria: Some Complexity Considerations,” *Games and Economic Behavior*, 1, 80–93.

GOLDFARB, D. (1983): “Worst-Case Complexity of the Shadow Simplex Algorithm,” Report, Department of Industrial Engineering and Operations Research, Columbia University.

——— (1994): “On the Complexity of the Simplex Method,” in *Advances in Optimization and Numerical Analysis*, Mathematics and Its Applications, Vol. 275. Dordrecht: Kluwer, 25–38.

GRAHAM, R. L., D. E. KNUTH, AND O. PATASHNIK (1994): *Concrete Mathematics*. Reading, MA: Addison–Wesley.

GRÜNBAUM, B. (2003): *Convex Polytopes* (Second Ed.). New York: Springer-Verlag.

KLEE, V., AND P. KLEINSCHMIDT (1987): “The d-Step Conjecture and Its Relatives,” *Mathematics of Operations Research*, 12, 718–755.

KLEE, V., AND G. J. MINTY (1972): “How Good Is the Simplex Algorithm?” in *Inequalities, III*, ed. by O. Shisha. New York: Academic Press, 159–175.

LEMKE, C. E. (1965): “Bimatrix Equilibrium Points and Mathematical Programming,” *Management Science*, 11, 681–689.

LEMKE, C. E., AND J. T. HOWSON, JR. (1964): “Equilibrium Points of Bimatrix Games,” *Journal of the Society for Industrial and Applied Mathematics*, 12, 413–423.

MCKELVEY, R. D., A. MCLENNAN, AND T. L. TUROCY (2005): “Gambit: Software Tools for Game Theory,” Version 0.2005.06.13. Available at http://econweb.tamu.edu/gambit.

MCLENNAN, A., AND R. TOURKY (2005): “From Imitation Games to Kakutani,” Unpublished Manuscript, University of Minnesota.

MEGIDDO, N. (1986): “On the Expected Number of Linear Complementarity Cones Intersected by Random and Semi-Random Rays,” *Mathematical Programming*, 35, 225–235.

MORRIS, W. D., JR. (1994): “Lemke Paths on Simple Polytopes,” *Mathematics of Operations Research*, 19, 780–789.

MOTWANI, R., AND P. RAGHAVAN (1995): *Randomized Algorithms*. Cambridge, U.K.: Cambridge University Press.

MURTY, K. G. (1978): “Computational Complexity of Complementary Pivot Methods,” *Mathematical Programming Study, 7: Complementary and Fixed Point Problems*. Amsterdam: North-Holland, 61–73.

——— (1980): “Computational Complexity of Parametric Linear Programming,” *Mathematical Programming*, 19, 213–219.

NASH, J. F. (1951): "Non-Cooperative Games," *The Annals of Mathematics*, 54, 286–295.

PAPADIMITRIOU, C. H. (1994): *Computational Complexity*. Reading, MA: Addison–Wesley.

——— (2001): "Algorithms, Games, and the Internet," in *Proceedings of the 33rd Symposium on the Theory of Computing*. New York: ACM Press, 749–753.

PORTER, R. W., E. NUDELMAN, AND Y. SHOHAM (2004): "Simple Search Methods for Finding a Nash Equilibrium," in *Proceedings of the 19th National Conference on Artificial Intelligence*. Cambridge, MA: AAAI/MIT Press, 664–669.

QUINT, T., AND M. SHUBIK (1997): "A Theorem on the Number of Nash Equilibria in a Bimatrix Game," *International Journal of Game Theory*, 26, 353–359.

SAVANI, R. (2004): "Challenge Instances for NASH," Research Report LSE-CDAM-2004–14, CDAM, London School of Economics.

——— (2005): "Finding Nash Equilibria of Bimatrix Games," Ph.D. Thesis, London School of Economics.

SAVANI, R., AND B. VON STENGEL (2004): "Exponentially Many Steps for Finding a Nash Equilibrium in a Bimatrix Game," *Research Report LSE-CDAM-2004-03*, CDAM, London School of Economics. [Extended abstract in *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. New York: IEEE Press, 258–267.]

SHAPLEY, L. S. (1974): "A Note on the Lemke–Howson Algorithm," *Mathematical Programming Study, 1: Pivoting and Extensions*, ed. by M. Balinski. Amsterdam: North-Holland, 175–189.

SMALE, S. (1983): "On the Average Number of Steps of the Simplex Method of Linear Programming," *Mathematical Programming*, 27, 241–262.

TODD, M. J. (2001): "The Many Facets of Linear Programming," *Mathematical Programming Series B*, 91, 417–436.

VON STENGEL, B. (1999): "New Maximal Numbers of Equilibria in Bimatrix Games," *Discrete and Computational Geometry*, 21, 557–568.

——— (2002): "Computing Equilibria for Two-Person Games," in *Handbook of Game Theory*, Vol. 3, ed. by R. J. Aumann and S. Hart. Amsterdam: North-Holland, 1723–1759.

VON STENGEL, B., A. H. VAN DEN ELZEN, AND A. J. J. TALMAN (2002): "Computing Normal Form Perfect Equilibria for Extensive Two-Person Games," *Econometrica*, 70, 693–715.

VOROB'EV, N. N. (1958): "Equilibrium Points in Bimatrix Games," *Theory of Probability and Its Applications*, 3, 297–309.

ZIEGLER, G. M. (1995): *Lectures on Polytopes*. New York: Springer-Verlag.