COMPUTING EQUILIBRIA FOR TWO-PERSON GAMES

To appear in: *Handbook of Game Theory*, Vol. 3, eds. R. J. Aumann and S. Hart, North-Holland, Amsterdam.

BERNHARD VON STENGEL*

London School of Economics

May 20, 1999

Contents

* This work was supported by a Heisenberg grant from the Deutsche Forschungsgemeinschaft.

1. Introduction

Finding Nash equilibria of strategic form or extensive form games can be difficult and tedious. A computer program for this task would allow greater detail of game-theoretic models, and enhance their applicability. Algorithms for solving games have been studied since the beginnings of game theory, and have proven useful for other problems in mathematical optimization, like linear complementarity problems.

This paper is a survey and exposition of *linear* methods for finding Nash equilibria. Above all, these apply to games with two players. In an equilibrium of a two-person game, the mixed strategy probabilities of one player equalize the expected payoffs for the pure strategies used by the other player. This defines an optimization problem with linear constraints. We do not consider nonlinear methods like simplicial subdivision for approximating fixed points, or systems of inequalities for higher-degree polynomials as they arise for noncooperative games with more than two players. These are surveyed in McKelvey and McLennan (1996).

First, we consider two-person games in strategic form (see also Parthasarathy and Raghavan, 1971; Raghavan, 1994, 1998). The classical algorithm by Lemke and Howson (1964) finds one equilibrium of a bimatrix game. It provides an elementary, constructive proof that such a game has an equilibrium, and shows that the number of equilibria is odd, except for degenerate cases. We follow Shapley's (1974) very intuitive geometric exposition of this algorithm. The maximization over linear payoff functions defines two *polyhedra* which provide further geometric insight. A complementary pivoting scheme describes the computation algebraically. Then we clarify the notion of *degeneracy*, which appears in the literature in various forms, most of which are equivalent. The lexicographic method extends pivoting algorithms to degenerate games. The problem of finding *all* equilibria of a bimatrix game can be phrased as a vertex enumeration problem for polytopes.

Second, we look at two methods for finding equilibria of strategic form games with additional refinement properties (see van Damme, 1987, 1998; Hillas and Kohlberg, 1998). Wilson (1992) modifies the Lemke–Howson algorithm for computing *simply stable* equilibria. These equilibria survive certain perturbations of the game that are easily represented by lexicographic methods for degeneracy resolution. Van den Elzen and Talman (1991) present a complementary pivoting method for finding a *perfect* equilibrium of a bimatrix game.

Third, we review methods for games in extensive form (see Hart, 1992). In principle, such game trees can be solved by converting them to the reduced strategic form and then applying the appropriate algorithms. However, this typically increases the size of the game description and the computation time exponentially, and is therefore infeasible. Approaches to avoiding this problem compute with a small fraction of the pure strategies, which are generated from the game tree as needed (Wilson, 1972; Koller and Megiddo, 1996). A strategic description of an extensive game that does not increase in size is the *sequence form.* The central idea, set forth independently by Romanovskii (1962), Selten (1988), Koller and Megiddo (1992), and von Stengel (1996a), is to consider only sequences of moves instead of pure strategies which are arbitrary combinations of moves. We will develop the problem of equilibrium computation for the strategic form in a way that can also be applied to the sequence form. In particular, the algorithm by van den Elzen and Talman (1991) for finding a perfect equilibrium carries over to the sequence form (von Stengel, van den Elzen and Talman, 1998).

The concluding section addresses issues of computational complexity, and mentions ongoing implementations of the algorithms.

2. Bimatrix games

We first introduce our notation, and recall notions from polytope theory and linear programming. Equilibria of a bimatrix game are the solutions to a linear complementarity problem. This problem is solved by the Lemke–Howson algorithm, which we explain in graph-theoretic, geometric, and algebraic terms. Then we consider degenerate games, and review enumeration methods.

2.1. Preliminaries

We use the following notation throughout. Let (A, B) be a bimatrix game, where A and B are $m \times n$ matrices of payoffs to the row player 1 and column player 2, respectively. All vectors are column vectors, so an m-vector x is treated as an $m \times 1$ matrix. A *mixed strategy* x for player 1 is a probability distribution on rows, written as an m-vector of probabilities. Similarly, a mixed strategy y for player 2 is an n-vector of probabilities for playing columns. The *support* of a mixed strategy is the set of pure strategies that have positive probability. A vector or matrix with all components zero is denoted $\mathbf{0}$. Inequalities like $x \geq \mathbf{0}$ between two vectors hold for all components. $B^\top$ is the matrix B transposed.

Let M be the set of the m pure strategies of player 1 and let N be the set of the n pure strategies of player 2. It is sometimes useful to assume that these sets are disjoint, as in

$$M = \{1, \ldots, m\}, \qquad N = \{m+1, \ldots, m+n\}. \tag{2.1}$$

Then $x \in \mathrm{I\!R}^M$ and $y \in \mathrm{I\!R}^N$, which means, in particular, that the components of y are y_j for $j \in N$. Similarly, the payoff matrices A and B belong to $\mathrm{I\!R}^{M \times N}$.

Denote the rows of A by a_i for $i \in M$, and the rows of $B^\top$ by b_j for $j \in N$ (so each $b_j^\top$ is a column of B). Then $a_i y$ is the expected payoff to player 1 for the pure strategy i when player 2 plays the mixed strategy y, and $b_j x$ is the expected payoff to player 2 for j when player 1 plays x.

A *best response* to the mixed strategy y of player 2 is a mixed strategy x of player 1 that maximizes his expected payoff $x^\top Ay$. Similarly, a best response y of player 2 to x maximizes her expected payoff $x^\top By$. A *Nash equilibrium* is a pair (x, y) of mixed strategies that are best responses to each other. Clearly, a mixed strategy is a best response to an opponent strategy if and only if it only plays pure strategies that are best responses with positive probability:

Theorem 2.1. *(Nash, 1951.) The mixed strategy pair (x, y) is a Nash equilibrium of (A, B) if and only if for all pure strategies i in M and j in N*

$$x_i > 0 \quad \Longrightarrow \quad a_i y = \max_{k \in M} \ a_k y \,, \tag{2.2}$$

$$y_j > 0 \quad \Longrightarrow \quad b_j x = \max_{k \in N} \ b_k x \,. \tag{2.3}$$

We recall some notions from the theory of (convex) polytopes (see Ziegler, 1995). An *affine combination* of points $z_1, \ldots, z_k$ in some Euclidean space is of the form $\sum_{i=1}^k z_i \lambda_i$ where $\lambda_1, \ldots, \lambda_k$ are reals with $\sum_{i=1}^k \lambda_i = 1$. It is called a *convex combination* if $\lambda_i \geq 0$ for all i. A set of points is *convex* if it is closed under forming convex combinations. Given points are *affinely independent* if none of these points is an affine combination of the others. A convex set has *dimension* d if and only if it has $d + 1$, but no more, affinely independent points.

A *polyhedron* P in $\mathbb{R}^d$ is a set $\{z \in \mathbb{R}^d \mid Cz \leq q\}$ for some matrix C and vector q. It is called *full-dimensional* if it has dimension d. It is called a *polytope* if it is bounded. A *face* of P is a set $\{\, z \in P \mid c^\top z = q_0\}$ for some $c \in \mathbb{R}^d$, $q_0 \in \mathbb{R}$ so that the inequality $c^\top z \leq q_0$ holds for all z in P. A *vertex* of P is the unique element of a 0-dimensional face of P. An *edge* of P is a one-dimensional face of P. A *facet* of a d-dimensional polyhedron P is a face of dimension $d - 1$. It can be shown that any nonempty face F of P can be obtained by turning some of the inequalities defining P into equalities, which are then called *binding* inequalities. That is, $F = \{\, z \in P \mid c_i z = q_i, \ i \in I\}$, where $c_i z \leq q_i$ for $i \in I$ are some of the rows in $Cz \leq q$. A facet is characterized by a single binding inequality which is *irredundant*, that is, the inequality cannot be omitted without changing the polyhedron (Ziegler, 1995, p. 72). A d-dimensional polyhedron P is called *simple* if no point belongs to more than d facets of P, which is true if there are no special dependencies between the facet-defining inequalities.

A *linear program* (LP) is the problem of maximizing a linear function over some polyhedron. The following notation is independent of the considered bimatrix game. Let M and N be finite sets, $I \subseteq M$, $J \subseteq N$, $A \in \mathbb{R}^{M \times N}$, $b \in \mathbb{R}^M$, $c \in \mathbb{R}^N$. Consider the polyhedron

$$
\begin{aligned}
P = \{\, x \in \mathbb{R}^N \mid \sum_{j\in N} a_{ij}x_j &= b_i, \quad i \in M - I,\\
\sum_{j\in N} a_{ij}x_j &\le b_i, \quad i \in I,\\
x_j &\ge 0, \quad j \in J \,\}.
\end{aligned}
$$

Any x belonging to P is called *primal feasible.* The *primal LP* is the problem

$$\text{maximize} \quad c^\top x \quad \text{subject to} \quad x \in P\,. \tag{2.4}$$

The corresponding *dual LP* has the feasible set

$$
\begin{aligned}
D = \{\, y \in \mathbb{R}^M \mid \sum_{i\in M} y_i a_{ij} &= c_j, \quad j \in N - J,\\
\sum_{i\in M} y_i a_{ij} &\ge c_j, \quad j \in J,\\
y_i &\ge 0, \quad i \in I \,\}
\end{aligned}
$$

and is the problem

$$\text{minimize} \quad y^\top b \quad \text{subject to} \quad y \in D\,. \tag{2.5}$$

Here the indices in I denote primal inequalities and corresponding nonnegative dual variables, whereas those in $M - I$ denote primal equality constraints and corresponding unconstrained dual variables. The sets J and $N - J$ play the same role with “primal” and “dual” interchanged. By reversing signs, the dual of the dual LP is again the primal. We recall the *duality theorem* of linear programming, which states (a) that for any primal and dual feasible solutions, the corresponding objective functions are mutual bounds, and (b) if the primal and the dual LP both have feasible solutions, then they have optimal solutions with the same value of their objective functions.

Theorem 2.2. *Consider the primal-dual pair of LPs (2.4), (2.5). Then*

(a) *(Weak duality.)* $c^\top x \le y^\top b$ *for all* $x \in P$ *and* $y \in D$.

(b) *(Strong duality.) If* $P \ne \emptyset$ *and* $D \ne \emptyset$ *then* $c^\top x = y^\top b$ *for some* $x \in P$ *and* $y \in D$.

For a proof see Schrijver (1986). As an introduction to linear programming we recommend Chvátal (1983).

2.2. Linear constraints and complementarity

Mixed strategies x and y of the two players are nonnegative vectors whose components sum up to one. These are linear constraints, which we define using

$$E = [1, \ldots, 1] \in \mathbb{R}^{1\times M}, \quad e = 1, \qquad F = [1, \ldots, 1] \in \mathbb{R}^{1\times N}, \quad f = 1\,. \tag{2.6}$$

Then the sets X and Y of mixed strategies are

$$X = \{\, x \in \mathbb{R}^M \mid Ex = e,\ x \geq \mathbf{0} \,\}, \qquad Y = \{\, y \in \mathbb{R}^N \mid Fy = f,\ y \geq \mathbf{0} \,\}\,. \tag{2.7}$$

With the extra notation in (2.6), the following considerations apply also if X and Y are more general polyhedra, where $Ex = e$ and $Fy = f$ may consist of more than a single row of equations. Such polyhedrally constrained games, first studied by Charnes (1953) for the zero-sum case, are useful for finding equilibria of extensive games (see Section 4).

Given a fixed y in Y, a best response of player 1 to y is a vector x in X that maximizes the expression $x^\top(Ay)$. That is, x is a solution to the LP

$$\text{maximize} \quad x^\top(Ay) \quad \text{subject to} \quad Ex = e, \quad x \geq \mathbf{0}. \tag{2.8}$$

The dual of this LP with variables u (by (2.6) only a single variable) states

$$\underset{u}{\text{minimize}} \quad e^\top u \quad \text{subject to} \quad E^\top u \geq Ay. \tag{2.9}$$

Both LPs are feasible. By Theorem 2.2(b), they have the same optimal value.

Consider now a *zero-sum game*, where $B = -A$. Player 2, when choosing y, has to assume that her opponent plays rationally and maximizes $x^\top Ay$. This maximum payoff to player 1 is the optimal value of the LP (2.8), which is equal to the optimal value $e^\top u$ of the dual LP (2.9). Player 2 is interested in minimizing $e^\top u$ by her choice of y. The constraints of (2.9) are linear in u and y even if y is treated as a variable, which must belong to Y. So a minmax strategy y of player 2 (minimizing the maximum amount she has to pay) is a solution to the LP

$$\underset{u,\,y}{\text{minimize}} \quad e^\top u \quad \text{subject to} \quad Fy = f, \quad E^\top u - Ay \geq \mathbf{0}, \quad y \geq \mathbf{0}. \tag{2.10}$$

Figure 2.1 shows an example.

The dual of the LP (2.10) has variables v and x corresponding to the primal constraints $Fy = f$ and $E^\top u - Ay \geq \mathbf{0}$, respectively. It has the form

$$\text{maximize} \quad f^\top v \quad \text{subject to} \quad Ex = e, \quad F^\top v - A^\top x \leq \mathbf{0}, \quad x \geq \mathbf{0}. \tag{2.11}$$

It is easy to verify that this LP describes the problem of finding a maxmin strategy x (with maxmin payoff $f^\top v$) for player 1. We have shown the following.

Theorem 2.3. *A zero-sum game with payoff matrix A for player 1 has the equilibrium (x, y) if and only if u, y is an optimal solution to the LP (2.10) and v, x is an optimal solution to its dual LP (2.11). Thereby, $e^\top u$ is the maxmin payoff to player 1 and $f^\top v$ is the minmax payoff to player 2. Both payoffs are equal and denote the value of the game.*

Thus, the "maxmin = minmax" theorem for zero-sum games follows directly from LP duality (see also Raghavan, 1994). This connection was noted by von

$$\begin{array}{c|cc|c|c}
 & \geq 0 & \geq 0 & & \\
u & y_4 & y_5 & & \\
\hline
0 & 1 & 1 & = & 1 \\
\hline
1 & 0 & -6 & & 0 \\
1 & -1 & -4 & \geq & 0 \\
1 & -3 & -3 & & 0 \\
\hline
\hline
1 & 0 & 0 & \to & \min
\end{array}
\qquad\qquad
\begin{array}{cc|c|cc||c}
 & v & 0 & 1 & 1 & 1 \\
\hline
\geq 0 & x_1 & 1 & 0 & -6 & 0 \\
\geq 0 & x_2 & 1 & -1 & -4 & 0 \\
\geq 0 & x_3 & 1 & -3 & -3 & 0 \\
\hline
 & & \| & \multicolumn{2}{c||}{\wedge\!|} & \downarrow \\
 & & 1 & 0 & 0 & \max
\end{array}$$

Figure 2.1. Left: Example of the LP (2.10) for a 3×2 zero-sum game. The objective function is separated by a line, nonnegative variables are marked by "≥ 0". Right: The dual LP (2.11), to be read vertically.

Neumann and Dantzig in the late 1940s when linear programming took its shape. Conversely, linear programs can be expressed as zero-sum games (see Dantzig, 1963, p. 277). There are standard algorithms for solving LPs, in particular Dantzig's Simplex algorithm. Usually, they compute a primal solution together with a dual solution which proves that the optimum is reached.

A best response x of player 1 against the mixed strategy y of player 2 is a solution to the LP (2.8). This is also useful for games that are not zero-sum. By strong duality, a feasible solution x is optimal if and only if there is a dual solution u fulfilling $E^\top u \geq Ay$ and $x^\top(Ay) = e^\top u$, that is, $x^\top(Ay) = (x^\top E^\top)u$ or equivalently

$$x^\top(E^\top u - Ay) = 0\,. \tag{2.12}$$

Because the vectors x and $E^\top u - Ay$ are nonnegative, (2.12) states that they have to be *complementary* in the sense that they cannot both have positive components in the same position. This characterization of an optimal primal-dual pair of feasible solutions is known as *complementary slackness* in linear programming. Since x has at least one positive component, the respective component of $E^\top u - Ay$ is zero and u is by (2.6) the maximum of the components of Ay. Any pure strategy i in M of player 1 is a best response to y if and only if the ith component of the slack vector $E^\top u - Ay$ is zero. That is, (2.12) is equivalent to (2.2).

For player 2, strategy y is a best response to x if and only if it maximizes $(x^\top B)y$ subject to $y \in Y$. The dual of this LP is the following analogous LP to (2.9): minimize $f^\top v$ subject to $F^\top v \geq B^\top x$. Here, a primal-dual pair y, v of feasible solutions is optimal if and only if, analogous to (2.12),

$$y^\top(F^\top v - B^\top x) = 0\,. \tag{2.13}$$

Considering these conditions for both players, this shows the following.

Theorem 2.4. *The game (A, B) has the Nash equilibrium (x, y) if and only if for suitable u, v*

$$\begin{array}{rrrrl} & & Ex & & = e \\ & & & Fy & = f \\ E^\top u & & & -\ Ay & \ge \mathbf{0} \\ & F^\top v & -\ B^\top x & & \ge \mathbf{0} \\ & & x\,, & y & \ge \mathbf{0} \end{array} \tag{2.14}$$

and (2.12), (2.13) hold.

The conditions in Theorem 2.4 define a so-called mixed *linear complementarity problem* (LCP). There are various solutions methods for LCPs. For a comprehensive treatment see Cottle, Pang, and Stone (1992). The most important method for finding one solution of the LCP in Theorem 2.4 is the Lemke–Howson algorithm.

2.3. The Lemke–Howson algorithm

In their seminal paper, Lemke and Howson (1964) describe an algorithm for finding one equilibrium of a bimatrix game. We follow Shapley's (1974) exposition of this algorithm. It requires disjoint pure strategy sets M and N of the two players as in (2.1). Any mixed strategy x in X and y in Y is *labeled* with certain elements of $M \cup N$. These labels denote the unplayed pure strategies of the player and the pure best responses of his or her opponent. For $i \in M$ and $j \in N$, let

$$\begin{aligned} X(i) &= \{x \in X \mid x_i = 0\,\}, \\ X(j) &= \{x \in X \mid b_j x \ge b_k x \ \text{ for all } k \in N\}, \\ Y(i) &= \{y \in Y \mid a_i y \ge a_k y \ \text{ for all } k \in M\}, \\ Y(j) &= \{y \in Y \mid y_j = 0\}. \end{aligned}$$

Then x has label k if $x \in X(k)$ and y has label k if $y \in Y(k)$, for $k \in M \cup N$. Clearly, the best response regions $X(j)$ for $j \in N$ are polytopes whose union is X. Similarly, Y is the union of the sets $Y(i)$ for $i \in M$. Then a Nash equilibrium is a *completely labeled* pair (x, y) since then by Theorem 2.1, any pure strategy k of a player is either a best response or played with probability zero, so it appears as a label of x or y.

Theorem 2.5. *A mixed strategy pair (x, y) in $X \times Y$ is a Nash equilibrium of (A, B) if and only if for all $k \in M \cup N$ either $x \in X(k)$ or $y \in Y(k)$ (or both).*

For the 3×2 bimatrix game (A, B) with

$$A = \begin{bmatrix} 0 & 6 \\ 2 & 5 \\ 3 & 3 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 4 & 3 \end{bmatrix}, \tag{2.15}$$

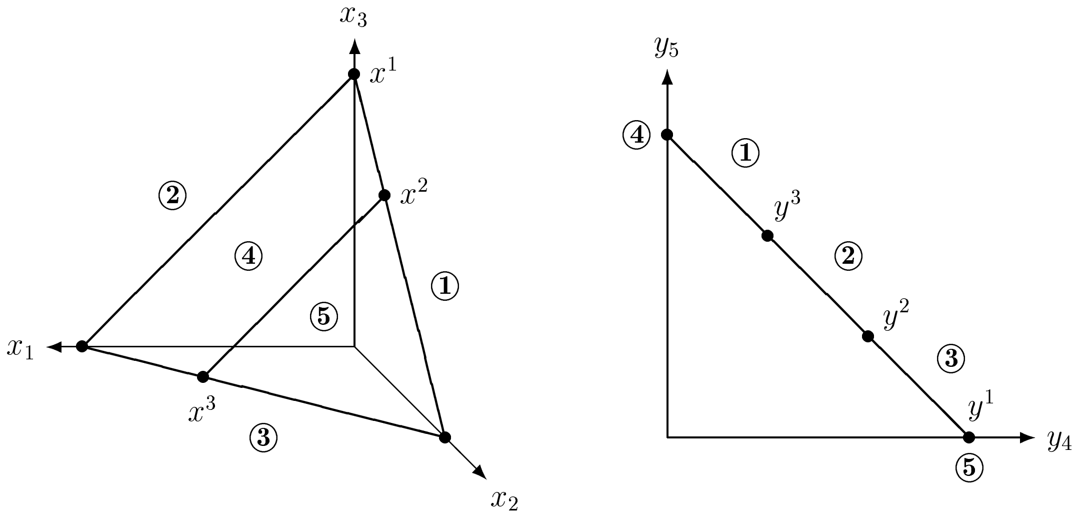


Figure 2.2. Mixed strategy sets X and Y of the players for the bimatrix game (A, B) in (2.15). The labels $1, 2, 3$, drawn as circled numbers, are the pure strategies of player 1 and marked in X where they have probability zero, in Y where they are best responses. The pure strategies of player 2 are similar labels $4, 5$. The dots mark points x and y with a maximum number of labels.

the labels of X and Y are shown in Figure 2.2. The equilibria are $(x^1, y^1) = \big((0,0,1)^\top, (1,0)^\top\big)$ where x^1 has the labels 1, 2, 4 (and y^1 the remaining labels 3 and 5), $(x^2, y^2) = \big((0, \frac{1}{3}, \frac{2}{3})^\top, (\frac{2}{3}, \frac{1}{3})^\top\big)$ with labels 1, 4, 5 for x^2, and $(x^3, y^3) = \big((\frac{2}{3}, \frac{1}{3}, 0)^\top, (\frac{1}{3}, \frac{2}{3})^\top\big)$ with labels 3, 4, 5 for x^3.

This geometric-qualitative inspection is very suitable for finding equilibria of games of size up to 3×3. It works by inspecting any point x in X with m labels and checking if there is a point y in Y having the remaining n labels. Usually, any x in X has at most m labels, and any y in Y has at most n labels. A game with this property is called *nondegenerate*, as stated in the following equivalent definition.

Definition 2.6. *A bimatrix game is called* nondegenerate *if the number of pure best responses to a mixed strategy never exceeds the size of its support.*

A game is usually nondegenerate since every additional label introduces an equation that reduces the dimension of the set of points having these labels by one. Then only single points x in X have m given labels and single points y in Y have n given labels, and no point has more labels. Nondegeneracy is discussed in greater detail in Section 2.6 below. Until further notice, we assume that the game is nondegenerate.

Theorem 2.7. *In a nondegenerate $m \times n$ bimatrix game (A, B), only finitely many points in X have m labels and only finitely many points in Y have n labels.*

Proof. Let K and L be subsets of $M \cup N$ with $|K| = m$ and $|L| = n$. There are only finitely many such sets. Consider the set of points in X having the labels in K, and the set of points in Y having the labels in L. By Theorem 2.10(c) below, these sets are empty or singletons. □

The finitely many points in the preceding theorem are used to define two graphs G_1 and G_2. Let G_1 be the graph whose vertices are those points x in X that have m labels, with an additional vertex $\mathbf{0}$ in $\mathbb{R}^M$ that has all labels i in M. Any two such vertices x and x' are joined by an edge if they differ in one label, that is, if they have $m-1$ labels in common. Similarly, let G_2 be the graph with vertices y in Y that have n labels, with the extra vertex $\mathbf{0}$ in $\mathbb{R}^N$ having all labels j in N, and edges joining those vertices that have $n-1$ labels in common. The *product graph* $G_1 \times G_2$ of G_1 and G_2 has vertices are (x, y) where x is a vertex of G_1, and y is a vertex of G_2. Its edges are given by $\{x\} \times \{y, y'\}$ for vertices x of G_1 and edges $\{y, y'\}$ of G_2, or by $\{x, x'\} \times \{y\}$ for edges $\{x, x'\}$ of G_1 and vertices y of G_2.

The Lemke–Howson algorithm can be defined combinatorially in terms of these graphs. Let $k \in M \cup N$, and call a vertex pair (x, y) of $G_1 \times G_2$ *k-almost completely labeled* if any l in $M \cup N - \{k\}$ is either a label of x or of y. Since two adjacent vertices x, x' in G_1, say, have $m-1$ labels in common, the edge $\{x, x'\} \times \{y\}$ of $G_1 \times G_2$ is also called k-almost completely labeled if y has the remaining n labels except k. The same applies to edges $\{x\} \times \{y, y'\}$ of $G_1 \times G_2$.

Then any equilibrium (x, y) is in $G_1 \times G_2$ adjacent to exactly one vertex pair (x', y') that is k-almost completely labeled: Namely, if k is the label of x, then x is joined to the vertex x' in G_1 sharing the remaining $m-1$ labels, and $y = y'$. If k is the label of y, then y is similarly joined to y' in G_2 and $x = x'$. In the same manner, a k-almost completely labeled pair (x, y) that is completely labeled has exactly two neighbors in $G_1 \times G_2$. These are obtained by dropping the unique duplicate label that x and y have in common, joining to an adjacent vertex either in G_1 and keeping y fixed, or in G_2 and keeping x fixed. This defines a unique k-almost completely labeled path in $G_1 \times G_2$ connecting one equilibrium to another. The algorithm is started from the *artificial* equilibrium $(\mathbf{0}, \mathbf{0})$ that has all labels, follows the path where label k is missing, and terminates at a Nash equilibrium of the game.

Figure 2.3 demonstrates this method for the above example. Let 2 be the missing label k. The algorithm starts with $x = (0, 0, 0)^\top$ and $y = (0, 0)^\top$. Step I: y stays fixed and x is changed in G_1 to $(0, 1, 0)^\top$, picking up label 5, which is now duplicate. Step II: dropping label 5 in G_2 changes y to $(0, 1)^\top$, picking up label 1. Step III: dropping label 1 in G_1 changes x to x^3, picking up label 4. Step IV: dropping label 4 in G_2 changes y to y^3 which has the missing label 2, terminating at the equilibrium (x^3, y^3). In a similar way, steps V and VI indicated in Figure 2.3

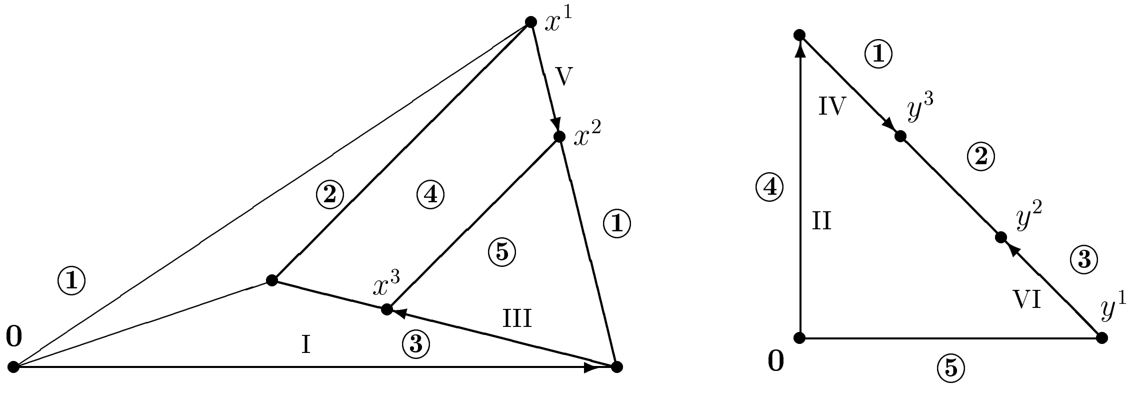


Figure 2.3. The graphs G_1 and G_2 for the game in (2.15). The set of 2-almost completely labeled pairs is formed by the paths with edges (in $G_1 \times G_2$) I–II–III–IV, connecting the artificial equilibrium $(\mathbf{0},\mathbf{0})$ and (x^3, y^3), and V–VI, connecting the equilibria (x^1, y^1) and (x^2, y^2).

join the equilibria (x^1, y^1) and (x^2, y^2) on a 2-almost completely labeled path. In general, one can show the following.

Theorem 2.8. *(Lemke and Howson, 1964; Shapley, 1974.) Let (A, B) be a nondegenerate bimatrix game and k be a label in $M \cup N$. Then the set of k-almost completely labeled vertices and edges in $G_1 \times G_2$ consists of disjoint paths and cycles. The endpoints of the paths are the equilibria of the game and the artificial equilibrium $(\mathbf{0},\mathbf{0})$. The number of Nash equilibria of the game is odd.*

This theorem provides a constructive, elementary proof that every nondegenerate game has an equilibrium, independently of the result of Nash (1951). By different labels k that are dropped initially, it may be possible to find different equilibria. However, this does not necessarily generate all equilibria, that is, the union of the k-almost completely labeled paths in Theorem 2.8 for all $k \in M \cup N$ may be disconnected (Shapley, 1974, p. 183, reports an example due to R. Wilson). For similar observations see Aggarwal (1973), Bastian (1976), Todd (1976, 1978). Shapley (1981) discusses more general methods as a potential way to overcome this problem.

2.4. Representation by polyhedra

The vertices and edges of the graphs G_1 and G_2 used in the definition of the Lemke–Howson algorithm can be represented as vertices and edges of certain polyhedra. Let

$$\begin{aligned} H_1 &= \{(x, v) \in \mathbb{R}^M \times \mathbb{R} \mid x \in X,\ B^\top x \le F^\top v\}, \\ H_2 &= \{(y, u) \in \mathbb{R}^N \times \mathbb{R} \mid y \in Y,\ Ay \le E^\top u\}. \end{aligned} \tag{2.16}$$

The elements of $H_1 \times H_2$ represent the solutions to (2.14). Figure 2.4 shows H_2 for the example (2.15). The horizontal plane contains Y as a subset. The scalar u, drawn vertically, is at least the maximum of the functions $a_i y$ for the rows a_i of A and for y in Y. The maximum itself shows which strategy of player 1 is a best response to y. Consequently, projecting H_2 to Y by mapping (y, u) to y, in Figure 2.4 shown as $(y, 0)$, reveals the subdivision of Y into best response regions $Y(i)$ for $i \in M$ as in Figure 2.2. Figure 2.4 shows also that the unbounded facets of H_2 project to the subsets $Y(j)$ of Y for $j \in N$. Furthermore, the maximally labeled points in Y marked by dots appear as projections of the vertices of H_2. Similarly, the facets of H_1 project to the subsets $X(k)$ of X for $k \in M \cup N$.

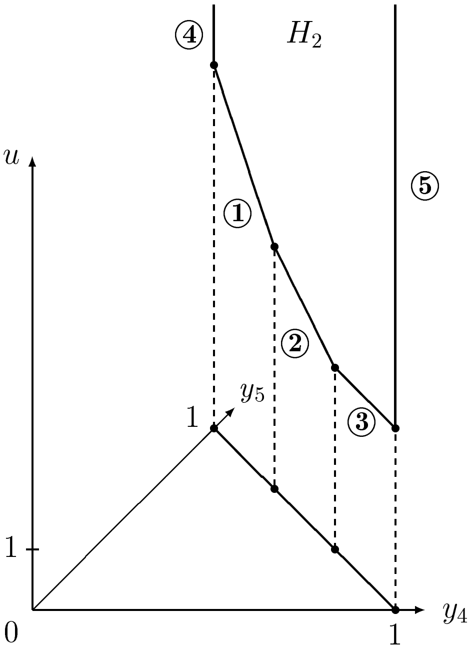


Figure 2.4. The polyhedron H_2 for the game in (2.15), and its projection to the set $\{ (y, 0) \mid (y, u) \in H_2 \}$. The vertical scale is displayed shorter. The circled numbers label the facets of H_2 analogous to Figure 2.2.

The graph structure of H_1 and H_2 with its vertices and edges is therefore identical to that of G_1 and G_2, except for the m unbounded edges of H_1 and the n unbounded edges of H_2 that connect to "infinity" rather than to the additional vertex $\mathbf{0}$ of G_1 and G_2, respectively.

The constraints (2.14) defining H_1 and H_2 can be simplified by eliminating the payoff variables u and v, which works if these are always positive. For that purpose, assume that

$$A \text{ and } B^\top \text{ are nonnegative and have no zero column.} \tag{2.17}$$

This assumption can be made without loss of generality since a constant can be added to all payoffs without changing the game in a material way, so that, for example, $A > \mathbf{0}$ and $B > \mathbf{0}$. For examples like (2.15), zero matrix entries are also admitted in (2.17). By (2.6), u and v are scalars and $E^\top$ and $F^\top$ are single columns with all components equal to one, which we denote by the vectors $\mathbf{1}_M$ in $\mathrm{I\!R}^M$ and $\mathbf{1}_N$ in $\mathrm{I\!R}^N$, respectively. Let

$$\begin{aligned} P_1 &= \{\, x' \in \mathrm{I\!R}^M \mid x' \geq \mathbf{0},\ B^\top x' \leq \mathbf{1}_N\}, \\ P_2 &= \{\, y' \in \mathrm{I\!R}^N \mid Ay' \leq \mathbf{1}_M,\ y' \geq \mathbf{0}\}\,. \end{aligned} \tag{2.18}$$

It is easy to see that (2.17) implies that P_1 and P_2 are full-dimensional polytopes, unlike H_1 and H_2.

The set H_1 is in one-to-one correspondence with $P_1 - \{\mathbf{0}\}$ with the map $(x, v) \mapsto x \cdot (1/v)$. Similarly, $(y, u) \mapsto y \cdot (1/u)$ defines a bijection $H_2 \to P_2 - \{\mathbf{0}\}$. These maps have the respective inverse functions $x' \mapsto (x, v)$ and $y' \mapsto (y, u)$ with

$$x = x' \cdot v, \quad v = 1/\mathbf{1}_M^\top x', \qquad y = y' \cdot u, \quad u = 1/\mathbf{1}_N^\top y'. \tag{2.19}$$

These bijections are not linear. However, they preserve the face incidences since a binding inequality in H_1 corresponds to a binding inequality in P_1 and vice versa. In particular, vertices have the same *labels* defined by the binding inequalities, which are some of the $m + n$ inequalities defining P_1 and P_2 in (2.18).

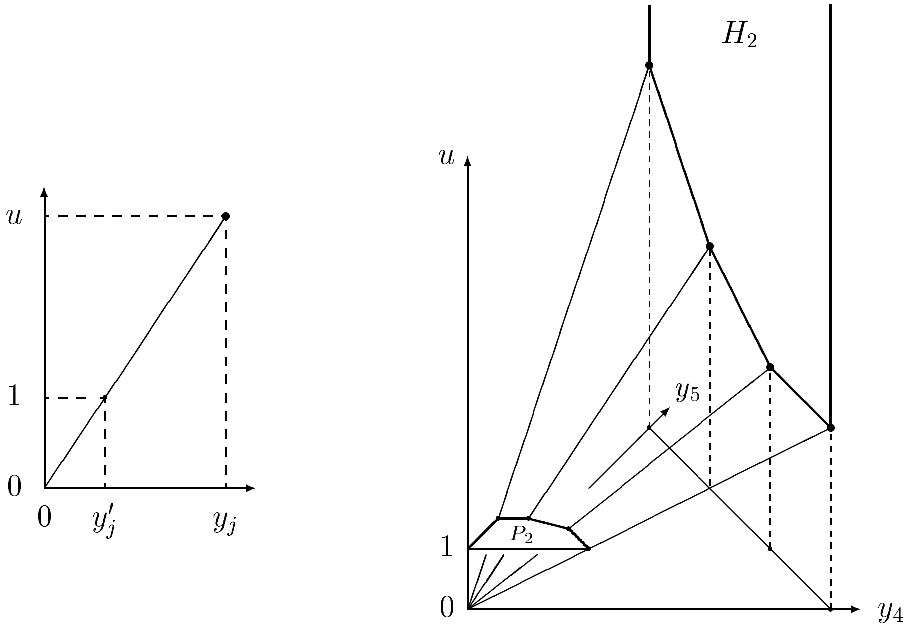


Figure 2.5. The map $H_2 \to P_2$, $(y, u) \mapsto y' = y \cdot (1/u)$ as a projective transformation with projection point $(\mathbf{0}, 0)$. The left hand side shows this for a single component y_j of y, the right hand side shows how P_2 arises in this way from H_2 in the example (2.15).

Figure 2.5 shows a geometric interpretation of the bijection $(y, u) \mapsto y \cdot (1/u)$ as a *projective transformation* (see Ziegler, 1995, Sect. 2.6). On the left hand side, the pair (y_j, u) is shown as part of (y, u) in H_2 for any component y_j of y. The line connecting this pair to $(0,0)$ contains the point $(y'_j, 1)$ with $y'_j = y_j/u$. Thus, $P_2 \times \{1\}$ is the intersection of the lines connecting any (y, u) in H_2 with $(\mathbf{0}, 0)$ in $\mathbb{R}^N \times \mathbb{R}$ with the set $\{(y', 1) \mid y' \in \mathbb{R}^N\}$. The vertices $\mathbf{0}$ of P_1 and P_2 do not arise as such projections, but correspond to H_1 and H_2 "at infinity".

2.5. Complementary pivoting

Traversing a polyhedron along its edges has a simple algebraic implementation known as *pivoting.* The constraints defining the polyhedron are thereby represented as linear equations with nonnegative variables. For $P_1 \times P_2$, these have the form

$$\begin{aligned} &Ay' + r &&= \mathbf{1}_M \\ B^\top x' & &&+ s = \mathbf{1}_N \end{aligned} \tag{2.20}$$

with $x', y', r, s \geq \mathbf{0}$ where $r \in \mathbb{R}^M$ and $s \in \mathbb{R}^N$ are vectors of *slack* variables. The system (2.20) is of the form

$$Cz = q \tag{2.21}$$

for a matrix C, right hand side q, and a vector z of nonnegative variables. The matrix C has full rank, so that q belongs always to the space spanned by the columns C_j of C. A *basis* β is given by a basis $\{C_j \mid j \in \beta\}$ of this column space, so that the square matrix C_β formed by these columns is invertible. The corresponding *basic solution* is the unique vector $z_\beta = (z_j)_{j\in\beta}$ with $C_\beta z_\beta = q$, where the variables z_j for j in β are called *basic variables*, and $z_j = 0$ for all *nonbasic* variables z_j, $j \notin \beta$, so that (2.21) holds. If this solution fulfills also $z \geq \mathbf{0}$, then the basis β is called *feasible.* If β is a basis for (2.21), then the corresponding basic solution can be read directly from the equivalent system $C_\beta^{-1}Cz = C_\beta^{-1}q$, called a *tableau*, since the columns of $C_\beta^{-1}C$ for the basic variables form the identity matrix. The tableau is equivalent to the system

$$z_\beta = C_\beta^{-1}q - \sum_{j\notin\beta} C_\beta^{-1}C_j z_j \tag{2.22}$$

which shows how the basic variables depend on the nonbasic variables.

Pivoting is a change of the basis where a nonbasic variable z_j for some j not in β *enters* and a basic variable z_i for some i in β *leaves* the set of basic variables. The pivot step is possible if and only if the coefficient of z_j in the ith row of the current tableau is nonzero, and is performed by solving the ith equation for z_j and then replacing z_j by the resulting expression in each of the remaining equations.

For a given entering variable z_j, the leaving variable is chosen to preserve feasibility of the basis. Let the components of $C_\beta^{-1}q$ be $\overline{q}_i$ and of $C_\beta^{-1}C_j$ be $\overline{c}_{ij}$, for $i \in \beta$. Then the largest value of z_j such that $z_\beta = C_\beta^{-1}q - C_\beta^{-1}C_j z_j \geq \mathbf{0}$ in (2.22) is obviously given by

$$\min\{\, \overline{q}_i/\overline{c}_{ij} \mid i \in \beta,\ \overline{c}_{ij} > 0\,\}. \tag{2.23}$$

This is called a *minimum ratio test.* Except in degenerate cases (see below), the minimum in (2.23) is unique and determines the leaving variable z_i uniquely. After pivoting, the new basis is $\beta \cup \{j\} - \{i\}$.

The choice of the entering variable depends on the solution that one wants to find. The Simplex method for linear programming is defined by pivoting with an entering variable that improves the value of the objective function. In the system (2.20), one looks for a *complementary* solution where

$$x'^{\top} r = 0, \qquad y'^{\top} s = 0 \tag{2.24}$$

because it implies with (2.19) the complementarity conditions (2.12) and (2.13) so that (x, y) is a Nash equilibrium by Theorem 2.4. In a basic solution to (2.20), every nonbasic variable has value zero and represents a binding inequality, that is, a facet of the polytope. Hence, each basis defines a vertex which is labeled with the indices of the nonbasic variables. The variables of the system come in *complementary pairs* (x_i, r_i) for the indices $i \in M$ and (y_j, s_j) for $j \in N$. Recall that the Lemke–Howson algorithm follows a path of solutions that have all labels in $M \cup N$ except for a missing label k. Thus a k-almost completely labeled vertex is a basis that has exactly one basic variable from each complementary pair, except for a pair of variables (x_k, r_k), say (if $k \in M$) that are both basic. Correspondingly, there is another pair of complementary variables that are both nonbasic, representing the duplicate label. One of them is chosen as the entering variable, depending on the direction of the computed path. The two possibilities represent the two k-almost completely labeled edges incident to that vertex. The algorithm is started with all components of r and s as basic variables and nonbasic variables $(x', y') = (\mathbf{0}, \mathbf{0})$. This initial solution fulfills (2.24) and represents the artificial equilibrium.

Algorithm 2.9. *(Complementary pivoting.) For a bimatrix game (A, B) fulfilling (2.17), compute a sequence of basic feasible solutions to the system (2.20) as follows.*

(a) *Initialize with basic variables $r = \mathbf{1}_M$, $s = \mathbf{1}_N$. Choose $k \in M \cup N$, and let the first entering variable be x'_k if $k \in M$ and y'_k if $k \in N$.*

(b) *Pivot such as to maintain feasibility using the minimum ratio test.*

(c) *If the variable z_i that has just left the basis has index k, stop. Then (2.24) holds and (x, y) defined by (2.19) is a Nash equilibrium. Otherwise, choose the complement of z_i as the next entering variable and go to* (b).

We demonstrate Algorithm 2.9 for the example (2.15). The initial basic solution in the form (2.22) is given by

$$\begin{aligned} r_1 &= 1 \phantom{{}-2y'_4} - 6y'_5 \\ r_2 &= 1 - 2y'_4 - 5y'_5 \\ r_3 &= 1 - 3y'_4 - 3y'_5 \end{aligned} \tag{2.25}$$

and

$$\begin{aligned} s_4 &= 1 - x'_1 \qquad\quad - 4x'_3 \\ s_5 &= 1 \qquad\quad - 2x'_2 - 3x'_3\,. \end{aligned} \tag{2.26}$$

Pivoting can be performed separately for these two systems since they have no variables in common. With the missing label 2 as in Figure 2.3, the first entering variable is x'_2. Then the second equation of (2.26) is rewritten as $x'_2 = \frac{1}{2} - \frac{3}{2}x'_3 - \frac{1}{2}s_5$ and s_5 leaves the basis. Next, the complement y'_5 of s_5 enters the basis. The minimum ratio (2.23) in (2.25) is 1/6, so that r_1 leaves the basis and (2.25) is replaced by the system

$$\begin{aligned} y'_5 &= \tfrac{1}{6} \qquad\quad - \tfrac{1}{6}r_1 \\ r_2 &= \tfrac{1}{6} - 2y'_4 + \tfrac{5}{6}r_1 \\ r_3 &= \tfrac{1}{2} - 3y'_4 + \tfrac{1}{2}r_1\,. \end{aligned} \tag{2.27}$$

Then the complement x'_1 of r_1 enters the basis and s_4 leaves, so that the system replacing (2.26) is now

$$\begin{aligned} x'_1 &= 1 - 4x'_3 - s_4 \\ x'_2 &= \tfrac{1}{2} - \tfrac{3}{2}x'_3 \qquad - \tfrac{1}{2}s_5\,. \end{aligned} \tag{2.28}$$

With y'_4 entering, the minimum ratio (2.23) in (2.27) is 1/12, where r_2 leaves the basis and (2.27) is replaced by

$$\begin{aligned} y'_5 &= \tfrac{1}{6} - \tfrac{1}{6}r_1 \\ y'_4 &= \tfrac{1}{12} + \tfrac{5}{12}r_1 - \tfrac{1}{2}r_2 \\ r_3 &= \tfrac{1}{4} - \tfrac{3}{4}r_1 - \tfrac{3}{2}r_2\,. \end{aligned} \tag{2.29}$$

Then the algorithm terminates since the variable r_2, with the missing label 2 as index, has become nonbasic. The solution defined by the final systems (2.28) and (2.29), with the nonbasic variables on the right hand side equal to zero, fulfills (2.24). Renormalizing x' and y' by (2.19) as probability vectors gives the equilibrium $(x, y) = (x^3, y^3)$ mentioned after (2.15) with payoffs 4 to player 1 and 2/3 to player 2.

Assumption (2.17) with the simple initial basis for the system (2.20) is used by Wilson (1992). Lemke and Howson (1964) assume $A < \mathbf{0}$ and $B < \mathbf{0}$, so that P_1 and P_2 are unbounded polyhedra and the almost completely labeled path starts at the vertex at the end of an unbounded edge. To avoid the renormalization (2.19), the Lemke–Howson algorithm can also be applied to the system (2.14) represented in equality form. Then the unconstrained variables u and v have no slack variables as counterparts and are always basic, so they never leave the basis and are disregarded in the minimum ratio test. Then the computation has the following *economic interpretation* (Wilson, 1992; van den Elzen, 1993): Let the missing label k belong to M. Then the basic slack variable r_k which is basic together with x_k can be interpreted as a "subsidy" payoff for the pure strategy k so that player 1 is in equilibrium. The algorithm terminates when that subsidy or the probability x_k vanishes. Player 2 is in equilibrium throughout the computation.

2.6. Degenerate games

The path computed by the Lemke–Howson algorithm is unique only if the game is nondegenerate. Like other pivoting methods, the algorithm can be extended to degenerate games by "lexicographic perturbation", as suggested by Lemke and Howson (1964). Before we explain this, we show that various definitions of nondegeneracy used in the literature are equivalent. In the following theorem, I_M denotes the identity matrix in $\mathbb{R}^{M\times M}$. Furthermore, a pure strategy i of player 1 is called *payoff equivalent* to a mixed strategy x of player 1 if it produces the same payoffs, that is, $a_i = x^\top A$. The strategy i is called *weakly dominated* by x if $a_i \le x^\top A$, and *strictly dominated* by x if $a_i < x^\top A$ holds. The same applies to strategies of player 2.

Theorem 2.10. *Let (A, B) be an $m\times n$ bimatrix game so that (2.17) holds. Then the following are equivalent.*

(a) *The game is nondegenerate according to Definition 2.6.*

(b) *For any x in X and y in Y, the rows of $\begin{bmatrix} I_M \\ B^\top \end{bmatrix}$ for the labels of x are linearly independent, and the rows of $\begin{bmatrix} A \\ I_N \end{bmatrix}$ for the labels of y are linearly independent.*

(c) *For any x in X with set of labels K and y in Y with set of labels L, the set $\bigcap_{k\in K} X(k)$ has dimension $m-|K|$, and the set $\bigcap_{l\in L} Y(l)$ has dimension $n-|L|$.*

(d) *P_1 and P_2 in (2.18) are simple polytopes, and any pure strategy of a player that is weakly dominated by or payoff equivalent to another mixed strategy is strictly dominated by some mixed strategy.*

(e) *In any basic feasible solution to (2.20), all basic variables have positive values.*

Lemke and Howson (1964) define nondegenerate games by condition (b). Krohn et al. (1991), and, in slightly weaker form, Shapley (1974), define nondegeneracy as in (c). Van Damme (1987, p. 52) has observed the implication (b)⇒(a). Some of the implications between the conditions (a)–(e) in Theorem 2.10 are easy to prove, whereas others require more work. For details of the proof see von Stengel (1996b).

The $m+n$ rows of the matrices in (b) define the inequalities for the polytopes P_1 and P_2 in (2.18), where the labels denote binding inequalities. This condition explains why a *generic* bimatrix game is nondegenerate with probability one: We call a game generic if each payoff is drawn randomly and independently from a continuous distribution, for example the normal distribution with small variance around an approximate value for the respective payoff. Then the rows of the matrices described in 2.10(b) are linearly independent with probability one, since a linear dependence imposes an equation on at least one payoff, which is fulfilled with probability zero. However, the strategic form of an extensive game (like Figure 4.1 below) is often degenerate since its payoff entries are not independent. A systematic treatment of degeneracy is therefore of interest.

The dimensionality condition in Theorem 2.10(c) has been explained informally before Theorem 2.7 above. The geometric interpretation of nondegeneracy in 2.10(d) consists of two parts. The polytope P_1 (and similarly P_2) is simple since a point that belongs to more than m facets of P_1 has too many labels. In the game

$$A = \begin{bmatrix} 0 & 6 \\ 2 & 5 \\ 3 & 3 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 4 & 4 \end{bmatrix}, \tag{2.30}$$

the polytope P_1 is not simple because its vertex $(0, 0, \frac{1}{4})^\top$ belongs to four facets. This game is degenerate since the pure strategy 3 of player 1 has two best responses. Apart from this, degeneracy may result due to a redundancy of the *description* of the polytope by inequalities (for example, if A has two identical rows of payoffs to player 1). It is not hard to show that such redundant inequalities correspond to weakly dominated strategies. A binding inequality of this sort defines a face of the respective polytope. The strict dominance in (d) asserts that this face is empty if the game is nondegenerate.

Theorem 2.10(e) states that every feasible *basis* of the system is *nondegenerate*, that is, all basic variables have positive values. This condition implies that the leaving variable in step (b) of Algorithm 2.9 is unique, since otherwise, another variable that could also leave the basis but stays basic will have value zero after the pivoting step. This concludes our remarks on Theorem 2.10.

The *lexicographic method* extends the minimum ratio test in such a way that the leaving variable is always unique, even in degenerate cases. The method simulates an infinitesimal perturbation of the right hand side of the given linear system (2.21), $z \geq \mathbf{0}$, and works as follows. Let Q be a matrix of full row rank with k columns. For any $\varepsilon \geq 0$, consider the system

$$Cz = q + Q \cdot (\varepsilon^1, \ldots, \varepsilon^k)^\top \tag{2.31}$$

which is equal to (2.21) for $\varepsilon = 0$ and which is a *perturbed* system for $\varepsilon > 0$. Let β be a basis for this system with basic solution

$$z_\beta = C_\beta^{-1} q + C_\beta^{-1} Q \cdot (\varepsilon^1, \ldots, \varepsilon^k)^\top = \overline{q} + \overline{Q} \cdot (\varepsilon^1, \ldots, \varepsilon^k)^\top \tag{2.32}$$

and $z_j = 0$ for $j \notin \beta$. It is easy to see that z_β is positive for all sufficiently small ε if and only if all rows of the matrix $[\overline{q}, \overline{Q}]$ are *lexico-positive*, that is, the first nonzero component of each row is positive. Then β is called a *lexico-feasible* basis. This holds in particular for $\overline{q} > \mathbf{0}$ when β is a nondegenerate basis for the unperturbed system. Because Q has full row rank, $\overline{Q}$ has no zero row, which implies that any feasible basis for the perturbed system is nondegenerate.

In consequence, the leaving variable for the perturbed system is always unique. It is determined by the following *lexico-minimum ratio test.* As for the minimum ratio test (2.23), let, for $i \in \beta$, the entries of the entering column $C_\beta^{-1}C_j$ be $\overline{c}_{ij}$, those of $\overline{q}$ in (2.32) be $\overline{q}_{i0}$, and those of $\overline{Q}$ be $\overline{q}_{il}$ for $1 \le l \le k$. Then the leaving variable is determined by the maximum choice of the entering variable z_j such that all basic variables z_i in (2.31) stay nonnegative, that is,

$$z_i = \overline{q}_{i0} + \overline{q}_{i1}\varepsilon^1 + \cdots + \overline{q}_{ik}\varepsilon^k - \overline{c}_{ij}z_j \ge 0$$

for all $i \in \beta$. For sufficiently small ε, the sharpest bound for z_j is obtained for that i in β with the *lexicographically smallest* row vector $1/\overline{c}_{ij} \cdot (\overline{q}_{i0}, \overline{q}_{i1}, \ldots, \overline{q}_{ik})$ where $\overline{c}_{ij} > 0$ (a vector is called lexicographically smaller than another if it is smaller in the first component where the vectors differ). No two of these row vectors are equal since $\overline{Q}$ has full row rank. Therefore, this lexico-minimum ratio test, which extends (2.23), determines the leaving variable z_i uniquely. By construction, it preserves the invariant that all computed bases are lexico-feasible, provided this holds for the initial basis like that in Algorithm 2.9(a) which is nondegenerate. Since the computed sequence of bases is unique, the computation cannot cycle and terminates like in the nondegenerate case.

The lexico-minimum ratio test can be performed without actually perturbing the system, since it only depends on the current basis β and Q in (2.32). The actual values of the basic variables are given by $\overline{q}$, which may have zero entries, so the perturbation applies as if ε is vanishing. The lexicographic method requires little extra work (and none for a nondegenerate game) since Q can be equal to C or to that part of C containing the identity matrix, so that $\overline{Q}$ in (2.32) is just the respective part of the current tableau. Wilson (1992) uses this to compute equilibria with additional stability properties, as discussed in Section 3.1 below.

2.7. Equilibrium enumeration and other methods

For a given bimatrix game, the Lemke–Howson algorithm finds at least one equilibrium. Sometimes, one wishes to find all equilibria, for example in order to know if an equilibrium is unique. A simple approach (as used by Dickhaut and Kaplan, 1991) is to enumerate all possible equilibrium supports, solve the corresponding linear equations for mixed strategy probabilities, and check if the unplayed pure strategies have smaller payoffs. In a nondegenerate game, both players use the same number of pure strategies in equilibrium, so only supports of equal cardinality need to be examined. They can be represented as $M \cap S$ and $N - S$ for any n-element subset S of $M \cup N$ except N. There are $\binom{m+n}{n} - 1$ many possibilities for S, which is exponential in the smaller dimension m or n of the bimatrix game. Stirling's asymptotic formula $\sqrt{2\pi n}(n/e)^n$ for the factorial $n!$ shows that in a square bimatrix game where $m = n$, the binomial coefficient $\binom{2n}{n}$ is asymptotically $4^n/\sqrt{\pi n}$. The number of equal sized supports is here not substantially smaller than the number 4^n of all possible supports.

An alternative is to inspect the vertices of $H_1 \times H_2$ defined in (2.16) if they represent equilibria. Mangasarian (1964) does this by checking if the bilinear function $x^\top(A+B)y-u-v$ has a maximum, that is, has value zero, so this is equivalent to the complementarity conditions (2.12) and (2.13). It is easier to enumerate the vertices of P_1 and P_2 in (2.18) since these are polytopes if (2.17) holds. Analogous to Theorem 2.5, a pair (x', y') in $P_1 \times P_2$, except $(\mathbf{0}, \mathbf{0})$, defines a Nash equilibrium (x, y) by (2.19) if it is completely labeled. The labels can be assigned directly to (x', y') as the binding inequalities. That is, (x', y') in $P_1 \times P_2$ has label i in M if $x'_i = 0$ or $a_i y = 1$, and label j in N if $b_j x' = 1$ or $y'_j = 0$ holds.

Theorem 2.11. *Let (A, B) be a bimatrix game so that (2.17) holds, and let V_1 and V_2 be the sets of vertices of P_1 and P_2 in (2.18), respectively. Then if (A, B) is nondegenerate, (x, y) given by (2.19) is a Nash equilibrium of (A, B) if and only if (x', y') is a completely labeled vertex pair in $V_1 \times V_2 - \{(\mathbf{0}, \mathbf{0})\}$.*

Thus, computing the vertex sets V_1 of P_1 and V_2 of P_2 and checking their labels finds all Nash equilibria of a nondegenerate game. This method was first suggested by Vorob'ev (1958), and later simplified by Kuhn (1961). An efficient and simple method for vertex enumeration is due to Avis and Fukuda (1992), which has apparently not yet been applied to bimatrix games.

The number of vertices of a polytope is in general exponential in the dimension. The maximal number is described in the following theorem, where $\lfloor t \rfloor$ for a real number t denotes the largest integer not exceeding t.

Theorem 2.12. *(Upper bound theorem for polytopes, McMullen, 1970.) The maximum number of vertices of a d-dimensional polytope with k facets is*

$$\Phi(d, k) = \binom{k - \lfloor \frac{d-1}{2} \rfloor - 1}{\lfloor \frac{d}{2} \rfloor} + \binom{k - \lfloor \frac{d}{2} \rfloor - 1}{\lfloor \frac{d-1}{2} \rfloor}.$$

For a self-contained proof of this theorem see Mulmuley (1994). This result shows that P_1 has at most $\Phi(m, n+m)$ and P_2 has at most $\Phi(n, m+n)$ vertices, including $\mathbf{0}$ which is not part of an equilibrium. In a nondegenerate game, any vertex is part of at most one equilibrium, so the smaller number of vertices of the polytope P_1 or P_2 is a bound for the number of equilibria.

Corollary 2.13. *(Keiding, 1997.) A nondegenerate $m \times n$ bimatrix game has at most* $\min\{\Phi(m, n+m), \Phi(n, m+n)\} - 1$ *equilibria.*

It is not hard to show that $m < n$ implies $\Phi(m, n+m) < \Phi(n, m+n)$. For $m = n$, Stirling's formula shows that $\Phi(n, 2n)$ is asymptotically $c \cdot (27/4)^{n/2}/\sqrt{n}$ or about $c \cdot 2.598^n/\sqrt{n}$, where the constant c is equal to $2\sqrt{2/3\pi}$ or about .921 if n is even, and $\sqrt{2/\pi}$ or about .798 if n is odd. Since 2.598^n grows less rapidly than 4^n, vertex enumeration is more efficient than support enumeration.

Although the upper bound in Corollary 2.13 is probably not tight, it is possible to construct bimatrix games that have a large number of Nash equilibria. The $n \times n$ bimatrix game where A and B are equal to the identity matrix has $2^n - 1$ Nash equilibria. Then both P_1 and P_2 are equal to the n-dimensional unit cube, where each vertex is part of a completely labeled pair. Quint and Shubik (1997) conjectured that no nondegenerate $n \times n$ bimatrix game has more equilibria. This follows from Corollary 2.13 for $n \leq 3$ and is shown for $n = 4$ by Keiding (1997) and McLennan and Park (1999). However, there are counterexamples for $n \geq 6$, with asymptotically $c \cdot (1+\sqrt{2})^n/\sqrt{n}$ or about $c \cdot 2.414^n/\sqrt{n}$ many equilibria, where c is $2^{3/4}/\sqrt{\pi}$ or about .949 if n is even, and $(2^{9/4} - 2^{7/4})/\sqrt{\pi}$ or about .786 if n is odd (von Stengel, 1999). These games are constructed with the help of polytopes which have the maximum number $\Phi(n, 2n)$ of vertices. This result suggests that vertex enumeration is indeed the appropriate method for finding all Nash equilibria.

For degenerate bimatrix games, Theorem 2.10(d) shows that P_1 or P_2 may be not simple. Then there may be equilibria (x, y) corresponding to completely labeled points (x', y') in $P_1 \times P_2$ where, for example, x' has more than m labels and y' has fewer than n labels and is therefore not a vertex of P_2. However, any such equilibrium is the convex combination of equilibria that are represented by vertex pairs, as shown by Mangasarian (1964). The set of Nash equilibria of an arbitrary bimatrix game is characterized as follows.

Theorem 2.14. *(Winkels, 1979; Jansen, 1981.) Let (A, B) be a bimatrix game so that (2.17) holds, let V_1 and V_2 be the sets of vertices of P_1 and P_2 in (2.18), respectively, and let R be the set of completely labeled vertex pairs in $V_1 \times V_2 - \{(\mathbf{0}, \mathbf{0})\}$. Then (x, y) given by (2.19) is a Nash equilibrium of (A, B) if and only if (x', y') belongs to the convex hull of some subset of R of the form $U_1 \times U_2$ where $U_1 \subseteq V_1$ and $U_2 \subseteq V_2$.*

Proof. Labels are preserved under convex combinations. Hence, if the set $U_1 \times U_2$ is contained in R, then any convex combination of its elements is also a completely labeled pair (x', y') that defines a Nash equilibrium by (2.19).

Conversely, assume (x', y') in $P_1 \times P_2$ corresponds to a Nash equilibrium of the game via (2.19). Let $I = \{\, i \in M \mid a_i y' < 1 \,\}$ and $J = \{\, j \in N \mid y'_j > 0 \,\}$, that is, x' has at least the labels in $I \cup J$. Then the elements z in P_1 fulfilling $z_i = 0$ for $i \in I$ and $b_j z = 1$ for $j \in J$ form a face of P_1 (defined by the sum of these equations, for example) which contains x'. This face is a polytope and therefore equal to the convex hull of its vertices, which are all vertices of P_1. Hence, x' is the positive convex combination $\sum_{k \in K} x^k \lambda_k$ of certain vertices x^k of P_1, where $\lambda_k > 0$

for $k \in K$. Similarly, y' is the positive convex combination $\sum_{l\in L} y^l \mu_l$ of certain vertices y^l of P_2, where $\mu_l > 0$ for $l \in L$. This implies the convex representation

$$(x', y') = \sum_{k\in K,\ l\in L} \lambda_k \mu_l (x^k, y^l)\,.$$

With $U_1 = \{\, x^k \mid k \in K\}$ and $U_2 = \{\, y^l \mid l \in L\}$, it remains to show $(x^k, y^l) \in G$ for all $k \in K$ and $l \in L$. Suppose otherwise that some (x^k, y^l) was not completely labeled, with some missing label, say $j \in N$, so that $b_j x^k < 1$ and $y^l_j > 0$. But then $b_j x' < 1$ since $\lambda_k > 0$ and $y'_j > 0$ since $\mu^l > 0$, so label j would also be missing from (x', y') contrary to the assumption. So indeed $U_1 \times U_2 \subseteq G$. □

The set R in Theorem 2.14 can be viewed as a bipartite graph with the completely labeled vertex pairs as edges. The subsets $U_1 \times U_2$ are *cliques* of this graph. The convex hulls of the maximal cliques of R are the *convex components* of Nash equilibria. Their union is the set of all equilibria, but they are not necessarily disjoint. The *topological* equilibrium components are unions of non-disjoint convex components.

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$$

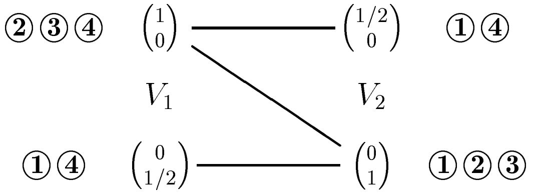


Figure 2.6. A game (A, B), and its set R of completely labeled vertex pairs in Theorem 2.14 as a bipartite graph. The labels denoting the binding inequalities in P_1 and P_2 are also shown for illustration.

An example is shown in Figure 2.6, where the convex components of Nash equilibria are, as sets of mixed strategies, $\{(1,0)^\top\} \times Y$ and $X \times \{(0,1)^\top\}$. This degenerate game illustrates the second part of condition 2.10(d): The polytopes P_1 and P_2 are simple but have vertices with more labels than the dimension due to weakly but not strongly dominated strategies. Dominated strategies could be iteratively eliminated, but this may not be desired here since the order of elimination matters. Knuth, Papadimitriou, and Tsitsiklis (1988) study computational aspects of strategy elimination where they overlook this fact; see also Gilboa, Kalai, and Zemel (1990, 1993).

Quadratic optimization is used for computing equilibria by Mills (1960), Mangasarian and Stone (1964), and Mukhamediev (1978). Audet et al. (1996) enumerate equilibria with a search over polyhedra defined by parameterized linear programs. Bomze (1992) describes an enumeration of the *evolutionarily stable* equilibria of a symmetric bimatrix game. Yanovskaya (1968), Howson (1972), Eaves (1973), and Howson and Rosenthal (1974) apply complementary pivoting to *polymatrix games*, which are multi-player games obtained as sums of pairwise interactions of the players.

3. Equilibrium refinements

Nash equilibria of a noncooperative game are not necessarily unique. A large number of *refinement* concepts have been invented for selecting some equilibria as more "reasonable" than others. We give an exposition (with further details in von Stengel, 1996b) of two methods that find equilibria with additional refinement properties. Wilson (1992) extends the Lemke–Howson algorithm so that it computes a *simply stable* equilibrium. A complementary pivoting method that finds a *perfect* equilibrium is due to van den Elzen and Talman (1991).

3.1. Simply stable equilibria

Kohlberg and Mertens (1986) define strategic *stability* of equilibria. Basically, a set of equilibria is called stable if every game nearby has equilibria nearby (Wilson, 1992). In degenerate games, certain equilibrium sets may not be stable. In the bimatrix game (A,B) in (2.30), for example, all convex combinations of (x^1,y^1) and (x^2,y^2) are equilibria, where $x^1=x^2=(0,0,1)^\top$ and $y^1=(0,1)^\top$ and $y^2=(\frac{1}{3},\frac{2}{3})^\top$. Another, isolated equilibrium is (x^3,y^3). As shown in the right picture of Figure 3.1, the first of these equilibrium sets is not stable since it disappears when the payoffs to player 2 for her second strategy 5 are slightly increased.

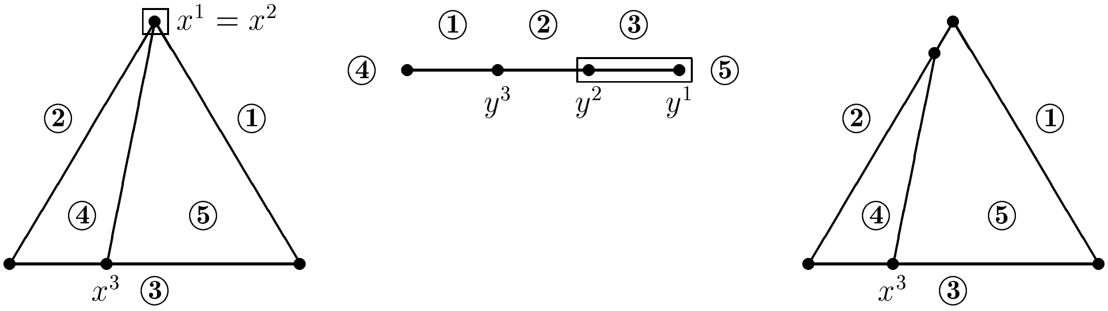


Figure 3.1. Left and center: Mixed strategy sets X and Y for the game (A,B) in (2.30) with labels similar to Figure 2.2. The game has an infinite set of equilibria indicated by the pair of rectangular boxes. Right: Mixed strategy set X where strategy 5 gets slightly higher payoffs, and only the equilibrium (x^3,y^3) remains.

Wilson (1992) describes an algorithm that computes a set of *simply stable* equilibria. There the game is not perturbed arbitrarily but only in certain systematic ways that are easily captured computationally. Simple stability is therefore weaker than the stability concepts of Kohlberg and Mertens (1986) and Mertens (1989, 1991). Simply stable sets may not be stable, but no such game has yet been found (Wilson, 1992, p. 1065). However, the algorithm is more efficient and seems practically useful compared to the exhaustive method by Mertens (1989).

The perturbations considered for simple stability do not apply to single payoffs but to pure strategies, in two ways. A *primal* perturbation introduces a small *minimum probability* for playing that strategy, even if it is not optimal. A *dual* perturbation introduces a small *bonus* for that strategy, that is, its payoff can be slightly smaller than the best payoff and yet the strategy is still considered optimal. In system (2.20), the variables x', y', r, s are perturbed by corresponding vectors ξ, η, ρ, σ that have small positive components, $\xi, \rho \in \mathrm{I\!R}^M$ and $\eta, \sigma \in \mathrm{I\!R}^N$. That is, (2.20) is replaced by

$$\begin{array}{lllll} & A(y'+\eta) + I_M(r+\rho) & & = \mathbf{1}_M \\ B^\top(x'+\xi) & & + I_N(s+\sigma) & = \mathbf{1}_N. \end{array} \tag{3.1}$$

If (3.1) and the complementarity condition (2.24) hold, then a variable x_i or y_j that is zero is replaced by ξ_i or η_j, respectively. After the transformation (2.19), these terms denote a small positive probability for playing the pure strategy i or j, respectively. So ξ and η represent primal perturbations.

Similarly, ρ and σ stand for dual perturbations. To see that ρ_i or σ_j indeed represents a bonus for i or j, respectively, consider the second set of equations in (3.1) with $\xi = \mathbf{0}$ for the example (2.30):

$$\begin{bmatrix} 1 & 0 & 4 \\ 0 & 2 & 4 \end{bmatrix} \begin{pmatrix} x_1' \\ x_2' \\ x_3' \end{pmatrix} + \begin{pmatrix} s_4+\sigma_4 \\ s_5+\sigma_5 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

If, say, $\sigma_5 > \sigma_4$, then one solution is $x_1' = x_2' = 0$ and $x_3' = (1-\sigma_5)/4$ with $s_5 = 0$ and $s_4 = \sigma_5 - \sigma_4 > 0$, which means that only the second strategy of player 2 is optimal, so the higher perturbation σ_5 represents a higher bonus for that strategy (as shown in the right picture in Figure 3.1). Dual perturbations are a generalization of primal perturbations, letting $\rho = A\eta$ and $\sigma = B^\top\xi$ in (3.1). Here, only special cases of these perturbations will be used, so it is useful to consider them both.

Denote the vector of perturbations in (3.1) by

$$(\xi, \eta, \rho, \sigma)^\top = \delta = (\delta_1, \ldots, \delta_k)^\top, \qquad k = 2(m+n). \tag{3.2}$$

For simple stability, Wilson (1992, p. 1059) considers only special cases of δ. For each $i \in \{1, \ldots, k\}$, the component δ_{i+1} (or δ_1 if $i = k$) represents the largest perturbation by some $\varepsilon > 0$. The subsequent components $\delta_{i+2}, \ldots, \delta_k, \delta_1, \ldots, \delta_i$ are equal to smaller perturbations $\varepsilon^2, \ldots, \varepsilon^k$. That is,

$$\begin{array}{rl} d_{i+j} = \varepsilon^j & \text{if } i+j \le k, \\ d_{i+j-k} = \varepsilon^j & \text{if } i+j > k, \end{array} \qquad 1 \le j \le k. \tag{3.3}$$

Definition 3.1. *(Wilson, 1992.) Let (A,B) be an $m\times n$ bimatrix game. Then a connected set of equilibria of (A,B) is called simply stable if for all $i=1,\ldots,k$, all sufficiently small $\varepsilon>0$, and (ξ,η,ρ,σ) as in (3.2), (3.3), there is a solution $r=(x',y',r,s)^\top\geq\mathbf{0}$ to (3.1) and (2.24) so that the corresponding strategy pair (x,y) defined by (2.19) is near that set.*

Due to the perturbation, (x,y) in Definition 3.1 is only an "approximate" equilibrium. When ε vanishes, then (x,y) becomes a member of the simply stable set. A perturbation with vanishing ε is mimicked by a lexico-minimum ratio test as described in Section 2.6 that extends step (b) of Algorithm 2.9. The perturbation (3.3) is therefore easily captured computationally. With (3.2), (3.3), the perturbed system (3.1) is of the form (2.31) with

$$z=(x',y',r,s)^\top,\quad C=\begin{bmatrix}\mathbf{0} & A & I_M & \mathbf{0}\\ B^\top & \mathbf{0} & \mathbf{0} & I_N\end{bmatrix},\quad q=\begin{bmatrix}\mathbf{1}_M\\ \mathbf{1}_N\end{bmatrix}\tag{3.4}$$

and $Q=[-C_{i+1},\ldots,-C_k,-C_1,\ldots,-C_i]$ if $C_1,\ldots,C_k$ are the columns of C. That is, Q is just $-C$ except for a cyclical shift of the columns, so that the lexico-minimum ratio test is easily performed using the current tableau.

The algorithm by Wilson (1992) computes a *path* of equilibria where all perturbations of the form (3.3) occur somewhere. Starting from the artificial equilibrium $(\mathbf{0},\mathbf{0})$, the Lemke–Howson algorithm is used to compute an equilibrium with a lexicographic order shifted by some i. Having reached that equilibrium, i is increased as long as the computed basic solution is lexico-feasible with that shifted order. If this is not possible for all i (as required for simple stability), a new Lemke–Howson path is started with the missing label determined by the maximally possible lexicographic shift. This requires several variants of pivoting steps. The final piece of the computed path represents the connected set in Definition 3.1.

3.2. Perfect equilibria and the tracing procedure

An equilibrium is *perfect* (Selten, 1975) if it is robust against certain small mistakes of the players. Mistakes are represented by small positive minimum probabilities for all pure strategies. We use the following characterization (Selten, 1975, p. 50, Theorem 7) as definition.

Definition 3.2. *(Selten, 1975.) An equilibrium (x,y) of a bimatrix game is called perfect if there is a continuous function $\varepsilon\mapsto(x(\varepsilon),y(\varepsilon))$ where $(x(\varepsilon),y(\varepsilon))$ is a pair of completely mixed strategies for all $\varepsilon>0$, $(x,y)=(x(0),y(0))$, and x is a best response to $y(\varepsilon)$ and y is a best response to $x(\varepsilon)$ for all ε.*

Positive minimum probabilities for all pure strategies define a special primal perturbation as considered for simply stable equilibria. Thus, as noted by Wilson (1992, p. 1042), his modification of the Lemke–Howson algorithm can also be used for computing a perfect equilibrium. Then it is not necessary to shift the lexicographic order, so the lexico-minimum ratio test described in Section 2.6 can be used with $Q = -C$.

Theorem 3.3. *Consider a bimatrix game (A, B) and, with (3.4), the LCP $Cz = q$, $z \geq \mathbf{0}$, (2.24). Then Algorithm 2.9, computing with bases β so that $C_\beta^{-1}[q, -C]$ is lexico-positive, terminates at a perfect equilibrium.*

Proof. Consider the computed solution to the LCP, which represents an equilibrium (x, y) by (2.19). The final basis β is lexico-positive, that is, for $Q = -C$ in the perturbed system (2.32), the basic variables z_β are all positive if $\varepsilon > 0$. In (2.32), replace $(\varepsilon, \ldots, \varepsilon^k)^\top$ by

$$\delta = (\xi, \eta, \rho, \sigma)^\top = (\varepsilon, \ldots, \varepsilon^{m+n}, 0, \ldots, 0)^\top, \tag{3.5}$$

so that z_β is still nonnegative. Then z_β contains the basic variables of the solution (x', y', r, s) to (3.1), with $\rho = \mathbf{0}$, $\sigma = \mathbf{0}$ by (3.5). This solution depends on ε, so $r = r(\varepsilon)$, $s = s(\varepsilon)$, and it determines the pair $x'(\varepsilon) = x' + \xi$, $y(\varepsilon) = y' + \eta$ which represents a completely mixed strategy pair if $\varepsilon > 0$. The computed equilibrium is equal to this pair for $\varepsilon = 0$, and it is a best response to this pair since it is complementary to the slack variables $r(\varepsilon), s(\varepsilon)$. Hence the equilibrium is perfect by Definition 3.2. □

A different approach to computing perfect equilibria of a bimatrix game is due to van den Elzen and Talman (1991, 1995); see also van den Elzen (1993). The method uses an arbitrary *starting point* (p, q) in the product $X \times Y$ of the two strategy spaces defined in (2.7). It computes a piecewise linear path in $X \times Y$ that starts at (p, q) and terminates at an equilibrium. The pair (p, q) is used throughout the computation as a reference point. The computation uses an auxiliary variable z_0, which can be regarded as a homotopy parameter. Initially, $z_0 = 1$. Then, z_0 is decreased and, after possible intermittent increases, eventually becomes zero, which terminates the algorithm.

The algorithm computes a sequence of basic solutions to the system

$$\begin{array}{rcccl} & & Ex & + & e\,z_0 = e \\ & & & Fy + & f\,z_0 = f \\ r = E^\top u & & & -\,Ay - & (Aq)z_0 \geq \mathbf{0} \\ s = & F^\top v - B^\top x & & - & (B^\top p)z_0 \geq \mathbf{0}\,, \\ & & x\,, & y\,, & z_0 \geq \mathbf{0}\,. \end{array} \tag{3.6}$$

These basic solutions contain at most one basic variable from each complementary pair (x_i, r_i) and (y_j, s_j) and therefore fulfill

$$x^\top r = 0, \qquad y^\top s = 0\,. \tag{3.7}$$

The constraints (3.6), (3.7) define an *augmented* LCP which differs from (2.14) only by the additional column for the variable z_0. That column is determined by (p, q). An initial solution is $z_0 = 1$ and $x = \mathbf{0}$, $y = \mathbf{0}$. As in Algorithm 2.9, the computation proceeds by complementary pivoting. It terminates when z_0 is zero and leaves the basis. Then the solution is an equilibrium by Theorem 2.4.

As observed in von Stengel, van den Elzen, and Talman (1998), the algorithm in this description is a special case of the algorithm by Lemke (1965) for solving an LCP (see also Murty, 1988; Cottle et al., 1992). Any solution to (3.6) fulfills $0 \leq z_0 \leq 1$, and the pair

$$(\overline{x}, \overline{y}) = (x + pz_0, y + qz_0) \tag{3.8}$$

belongs to $X \times Y$ since $Ep = e$ and $Fq = f$. Hence, $(\overline{x}, \overline{y})$ is a pair of mixed strategies, initially equal to the starting point (p, q). For $z_0 = 0$, it is the computed equilibrium. The set of these pairs $(\overline{x}, \overline{y})$ is the computed piecewise linear path in $X \times Y$. In particular, the computed solution is always bounded. The algorithm can therefore never encounter an unbounded ray of solutions, which in general may cause Lemke's algorithm to fail. The computed pivoting steps are unique by using lexicographic degeneracy resolution. This proves that the algorithm terminates.

In (3.8), the positive components x_i and y_j of x and y describe which pure strategies i and j, respectively, are played with higher probability than the minimum probabilities $p_i z_0$ and $q_j z_0$ as given by (p, q) and the current value of z_0. By the complementarity condition (3.7), these are *best responses* to the current strategy pair $(\overline{x}, \overline{y})$. Therefore, any point on the computed path is an *equilibrium* of the *restricted* game where each pure strategy has at least the probability it has under $(p, q) \cdot z_0$. Considering the final line segment of the computed path, one can therefore show the following.

Theorem 3.4. *(Van den Elzen and Talman, 1991.) Lemke's complementary pivoting algorithm applied to the augmented LCP (3.6), (3.7) terminates at a perfect equilibrium if the starting point (p, q) is completely mixed.*

As shown by van den Elzen and Talman (1995), their algorithm also emulates the *linear tracing procedure* of Harsanyi and Selten (1988). The tracing procedure is an adjustment process to arrive at an equilibrium of the game when starting from a prior (p, q). It traces a pair of strategy pairs $(\overline{x}, \overline{y})$. Each such pair is an equilibrium in a parameterized game where the prior is played with probability z_0 and the currently used strategies with probability $1 - z_0$. Initially, $z_0 = 1$ and the players react against the prior. Then they simultaneously and gradually adjust their expectations and react optimally against these revised expectations, until they reach an equilibrium of the original game.

4. Extensive form games

In a game in extensive form, successive moves of the players are represented by edges of a tree. The standard way to find an equilibrium of such a game has been to convert it to strategic form, where each combination of moves of a player is a strategy. However, this typically increases the description of the game exponentially. In order to reduce this complexity, Wilson (1972) and Koller and Megiddo (1996) describe computations that use mixed strategies with *small support.* A different approach uses the *sequence form* of the game where pure strategies are replaced by move sequences, which are small in number. We describe it following von Stengel (1996a), and mention similar work by Romanovskii (1962), Selten (1988), Koller and Megiddo (1992), and further developments.

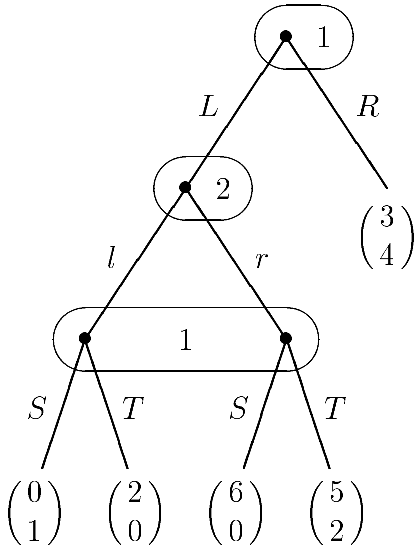
4.1. Extensive form and reduced strategic form

The basic structure of an extensive game is a finite tree. The nodes of the tree represent game states. The game starts at the root (initial node) of the tree and ends at a leaf (terminal node), where each player receives a payoff. The nonterminal nodes are called *decision nodes.* The player's *moves* are assigned to the outgoing edges of the decision node. The decision nodes are partitioned into *information sets*, introduced by Kuhn (1953). All nodes in an information set belong to the same player, and have the same moves. The interpretation is that when a player makes a move, he only knows the information set but not the particular node he is at. Some decision nodes may belong to *chance* where the next move is made according to a known probability distribution.

We denote the set of information sets of player i by H_i, information sets by h, and the set of moves at h by C_h. In the extensive game in Figure 4.1, moves are marked by upper case letters for player 1 and by lower case letters for player 2. Information sets are indicated by ovals. The two information sets of player 1 have move sets $\{L, R\}$ and $\{S, T\}$, and the information set of player 2 has move set $\{l, r\}$.

Equilibria of an extensive game can found recursively by considering *subgames* first. A subgame is a subtree of the game tree that includes all information sets containing a node of the subtree. In a game with *perfect information*, where every information set is a singleton, every node is the root of a subgame, so that an equilibrium can be found by backward induction. In games with imperfect information, equilibria of subgames are sometimes easy to find. Figure 4.1, for example, has a subgame starting at the decision node of player 2. It is equivalent to a 2×2 game and has a unique mixed equilibrium with probability $2/3$ for the moves S and r, respectively, and expected payoff 4 to player 1 and $2/3$ to player 2. Preceded by move L of player 1, this defines the unique *subgame perfect* equilibrium of the game.

In general, Nash equilibria of an extensive game (in particular one without subgames) are defined as equilibria of its *strategic form.* There, a *pure strategy*



$$A = \begin{array}{ccc|l} \emptyset & l & r & \\ \hline & & & \emptyset \\ & & & L \\ 3 & & & R \\ & 0 & 6 & LS \\ & 2 & 5 & LT \end{array} \qquad B = \begin{array}{ccc|l} \emptyset & l & r & \\ \hline & & & \emptyset \\ & & & L \\ 4 & & & R \\ & 1 & 0 & LS \\ & 0 & 2 & LT \end{array}$$

Figure 4.1. Left: A game in extensive form. Its reduced strategic form is (2.30). Right: The *sequence form* payoff matrices A and B. Rows and columns correspond to the sequences of the players which are marked at the side. Any sequence pair not leading to a leaf has matrix entry zero, which is left blank.

of player i prescribes a deterministic move at each information set, so it is an element of $\prod_{h \in H_i} C_h$. In Figure 4.1, the pure strategies of player 1 are the move combinations $\langle L, S\rangle$, $\langle L, T\rangle$, $\langle R, S\rangle$, and $\langle R, T\rangle$. In the *reduced strategic form*, moves at information sets that cannot be reached due to an earlier own move are identified. In Figure 4.1, this reduction yields the pure strategy (more precisely, equivalence class of pure strategies) $\langle R, *\rangle$, where $*$ denotes an arbitrary move. The two pure strategies of player 2 are her moves l and r. The reduced strategic form (A, B) of this game is then as in (2.30). This game is *degenerate* even if the payoffs in the extensive game are generic, because player 2 receives payoff 4 when player 1 chooses R (the bottom row of the bimatrix game) irrespective of her own move. Furthermore, the game has an equilibrium which is not subgame perfect, where player 1 chooses R and player 2 chooses l with probability at least 2/3.

A player may have *parallel* information sets that are not distinguished by own earlier moves. In particular, these arise when a player receives information about an earlier move by another player. Combinations of moves at parallel information sets cannot be reduced (see von Stengel, 1996b, for further details). This causes a multiplicative growth of the number of strategies even in the reduced strategic form. In general, the reduced strategic form is therefore *exponential* in the size of the game tree. Strategic form algorithms are then exceedingly slow except for very small game trees. Although extensive games are convenient modeling tools, their use has partly been limited for this reason (Lucas, 1972).

Wilson (1972) applies the Lemke–Howson algorithm to the strategic form of an extensive game while storing only those pure strategies that are actually played. That is, only the positive mixed strategy probabilities are computed explicitly. These correspond to basic variables x'_i or y'_j in Algorithm 2.9. The slack variables r_i and s_j are merely known to be nonnegative. For the pivoting step, the leaving variable is determined by a minimum ratio test which is performed *indirectly* for the tableau rows corresponding to basic slack variables. If, for example, y'_k enters the basis in step 2.9(b), then the conditions $y'_j \geq 0$ and $r_i \geq 0$ for the basic variables y_j and r_i determine the value of the entering variable by the minimum ratio test. In Wilson (1972), this test is first performed by ignoring the constraints $r_i \geq 0$, yielding a new mixed strategy y^0 of player 2. Against this strategy, a pure best response i of player 1 is computed from the game tree by a *subroutine*, essentially backward induction. If i has the same payoff as the currently used strategies of player 1, then $r \geq \mathbf{0}$ and some component of y leaves the basis. Otherwise, the payoff for i is higher and $r_i < 0$. Then at least the inequality $r_i \geq 0$ is violated, which is now added for a new minimum ratio test. This determines a new, smaller value for the entering variable and a corresponding mixed strategy y^1. Against this strategy, a best response is computed again. This process is repeated, computing a sequence of mixed strategies $y^0, y^1, \ldots, y^t$, until $r \geq \mathbf{0}$ holds and the correct leaving variable r_i is found.

Each pure strategy used in this method is stored explicitly as a tuple of moves. Their number should stay small during the computation. In the description by Wilson (1972) this is not guaranteed. However, the desired small support of the computed mixed strategies can be achieved by maintaining an additional system of linear equations for *realization weights* of the leaves of the game tree and with a *basis crashing* subroutine, as shown by Koller and Megiddo (1996).

The best response subroutine in Wilson's (1972) algorithm requires that the players have *perfect recall*, that is, all nodes in an information set of a player are preceded by the same earlier moves of that player (Kuhn, 1953). For finding *all* equilibria, Koller and Megiddo (1996) show how to enumerate small supports in a way that can also be applied to extensive games without perfect recall.

4.2. Sequence form

The use of pure strategies can be avoided altogether by using *sequences* of moves instead. The unique path from the root to any node of the tree defines a sequence of moves for player i. We assume player i has perfect recall. That is, any two nodes in an information set h in H_i define the same sequence for that player, which we denote by σ_h. Let S_i be the set of sequences of moves for player i. Then any σ in S_i is either the empty sequence $\emptyset$ or uniquely given by its last move c at the information set h in H_i, that is, $\sigma = \sigma_h c$. Hence, $S_i = \{\,\emptyset\,\} \cup \{\,\sigma_h c \mid h \in H_i,\; c \in C_h\,\}$. So player i does not have more sequences than the tree has nodes.

The *sequence form* of the extensive game, described in detail in von Stengel (1996), is similar to the strategic form but uses sequences instead of pure strategies, so it is a very compact description. Randomization over sequences is thereby described as follows.

A *behavior strategy* β of player i is given by probabilities $\beta(c)$ for his moves c which fulfill $\beta(c) \geq 0$ and $\sum_{c \in C_h} \beta(c) = 1$ for all h in H_i. This definition of β can be extended to the sequences σ in S_i by writing

$$\beta[\sigma] = \prod_{c \text{ in } \sigma} \beta(c). \tag{4.1}$$

A pure strategy π of player i can be regarded as a behavior strategy with $\pi(c) \in \{0, 1\}$ for all moves c. Thus, $\pi[\sigma] \in \{0, 1\}$ for all σ in S_i. The pure strategies π with $\pi[\sigma] = 1$ are those "agreeing" with σ by prescribing all the moves in σ, and arbitrary moves at the information sets not touched by σ.

A mixed strategy μ of player i assigns a probability $\mu(\pi)$ to every pure strategy π. In the sequence form, a randomized strategy of player i is described by the *realization probabilities* of playing the sequences σ in S_i. For a behavior strategy β, these are obviously $\beta[\sigma]$ as in (4.1). For a mixed strategy μ of player i, they are obtained by summing over all pure strategies π of player i, that is,

$$\mu[\sigma] = \sum_{\pi} \mu(\pi)\pi[\sigma]\,. \tag{4.2}$$

For player 1, this defines a map x from S_1 to $\mathbb{R}$ by $x(\sigma) = \mu[\sigma]$ for σ in S_1 which we call the *realization plan* of μ or a realization plan for player 1. A realization plan for player 2, similarly defined on S_2, is denoted y.

Theorem 4.1. *(Koller and Megiddo, 1992; von Stengel, 1996.) For player 1, x is the realization plan of a mixed strategy if and only if $x(\sigma) \geq 0$ for all $\sigma \in S_1$ and*

$$\begin{aligned} x(\emptyset) &= 1, \\ \sum_{c \in C_h} x(\sigma_h c) &= x(\sigma_h), \qquad h \in H_1. \end{aligned} \tag{4.3}$$

A realization plan y of player 2 is characterized analogously.

Proof. Equations (4.3) hold for the realization probabilities $x(\sigma) = \beta[\sigma]$ for a behavior strategy β and thus for every pure strategy π, and therefore for their convex combinations in (4.2) with the probabilities $\mu(\pi)$. □

To simplify notation, we write realization plans as vectors $x = (x_\sigma)_{\sigma\in S_1}$ and $y = (y_\sigma)_{\sigma\in S_2}$ with sequences as subscripts. According to Theorem 4.1, these vectors are characterized by

$$x \geq \mathbf{0}, \quad Ex = e, \qquad y \geq \mathbf{0}, \quad Fy = f \tag{4.4}$$

for suitable matrices E and F, and vectors e and f that are equal to $(1, 0 \ldots, 0)^\top$, where E and e have $1 + |H_1|$ rows and F and f have $1 + |H_2|$ rows. In Figure 4.1, the sets of sequences are $S_1 = \{\emptyset, L, R, LS, LT\}$ and $S_2 = \{\emptyset, l, r\}$, and in (4.4),

$$E = \begin{bmatrix} 1 & & & & \\ -1 & 1 & 1 & & \\ & & -1 & 1 & 1 \end{bmatrix}, \quad e = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad F = \begin{bmatrix} 1 & & \\ -1 & 1 & 1 \end{bmatrix}, \quad f = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

The number of information sets and therefore the number of rows of E and F is at most linear in the size of the game tree.

Mixed strategies of a player are called *realization equivalent* (Kuhn, 1953) if they define the same realization probabilities for all nodes of the tree, given any strategy of the other player. For reaching a node, only the players' sequences matter, which shows that the realization plan contains the strategically relevant information for playing a mixed strategy:

Theorem 4.2. *(Koller and Megiddo, 1992; von Stengel, 1996.) Two mixed strategies μ and μ' of player i are realization equivalent if and only if they have the same realization plan, that is, $\mu[\sigma] = \mu'[\sigma]$ for all $\sigma \in S_i$.*

Any realization plan x of player 1 (and similarly y for player 2) naturally defines a behavior strategy β where the probability for move c is $\beta(c) = x(\sigma_h c)/x(\sigma_h)$, and arbitrary, for example, $\beta(c) = 1/|C_h|$, if $x(\sigma_h) = 0$ since then h cannot be reached.

Corollary 4.3. *(Kuhn, 1953.) For a player with perfect recall, any mixed strategy is realization equivalent to a behavior strategy.*

In Theorem 4.2, a mixed strategy μ is mapped to its realization plan by regarding (4.2) as a linear map with given coefficients $\pi[\sigma]$ for the pure strategies π. This maps the simplex of mixed strategies of a player to the polytope of realization plans. These polytopes are characterized by (4.4) as asserted in Theorem 4.1. They define the player's *strategy spaces* in the sequence form, which we denote by X and Y as in (2.7). The vertices of X and Y are the players' pure strategies up to realization equivalence, which is the identification of pure strategies used in the reduced strategic form. However, the dimension and the number of facets of X and Y is reduced from exponential to linear size.

Sequence form *payoffs* are defined for pairs of sequences whenever these lead to a leaf, multiplied by the probabilities of chance moves on the path to the leaf. This defines two sparse matrices A and B of dimension $|S_1|\times|S_2|$ for player 1 and player 2, respectively. For the game in Figure 2.1, A and B are shown in Figure 4.1 on the right. When the players use the realization plans x and y, the expected payoffs are $x^\top Ay$ for player 1 and $x^\top By$ for player 2. These terms represent the sum over all leaves of the payoffs at leaves multiplied by their realization probabilities.

The formalism in Section 2.2 can be applied to the sequence form without change. For zero-sum games, one obtains the analogous result to Theorem 2.3. It was first proved by Romanovskii (1962). He constructs a constrained matrix game (see Charnes, 1953) which is equivalent to the sequence form. The perfect recall assumption is weakened by Yanovskaya (1970). Until recently, these publications were overlooked in the English-speaking community.

Theorem 4.4. *(Romanovskii, 1962; von Stengel, 1996.) The equilibria of a two-person zero-sum game in extensive form with perfect recall are the solutions of the LP (2.10) with sparse sequence form payoff matrix A and constraint matrices E and F in (4.4) defined by Theorem 4.1. The size of this LP is linear in the size of the game tree.*

Selten (1988, pp. 226, 237ff) defines sequence form strategy spaces and payoffs to exploit their linearity, but not for computational purposes. Koller and Megiddo (1992) describe the first polynomial-time algorithm for solving two-person zero-sum games in extensive form, apart from Romanovskii's result. They define the constraints (4.3) for playing sequences σ of a player with perfect recall. For the other player, they still consider pure strategies. This leads to an LP with a linear number of variables x_σ but possibly exponentially many inequalities. However, these can be evaluated as needed, similar to Wilson (1972). This solves efficiently the "separation problem" when using the ellipsoid method for linear programming.

For non-zero-sum games, the sequence form defines an LCP analogous to Theorem 2.4. Again, the point is that this LCP has the same size as the game tree. The Lemke–Howson algorithm cannot be applied to this LCP, since the missing label defines a single pure strategy, which would involve more than one sequence in the sequence form. Koller, Megiddo, and von Stengel (1996) describe how to use the more general complementary pivoting algorithm by Lemke (1965) for finding a solution to the LCP derived from the sequence form. This algorithm uses an additional variable z_0 and a corresponding column to augment the LCP. However, that column is just some positive vector, which requires a very technical proof that Lemke's algorithm terminates.

In von Stengel, van den Elzen, and Talman (1998), the augmented LCP (3.6), (3.7) is applied to the sequence form. The column for z_0 is derived from a starting pair (p, q) of realization plans. The computation has the interpretation described in Section 3.2. Similar to Theorem 3.4, the computed equilibrium can be shown to be strategic-form perfect if the starting point is completely mixed.

5. Computational issues

How long does it take to find an equilibrium of a bimatrix game? The Lemke–Howson algorithm has exponential running time for some specifically constructed, even zero-sum, games. However, this does not seem to be the typical case. In practice, numerical stability is more important (Tomlin, 1978; Cottle et al., 1992). Interior point methods that are provably polynomial as for linear programming are not known for LCPs arising from games; for other LCPs see Kojima et al. (1991). The computational complexity of finding one equilibrium is unclear. By Nash's theorem, an equilibrium exists, but the problem is to construct one. Megiddo (1988), Megiddo and Papadimitriou (1989), and Papadimitriou (1994) study the computational complexity of problems of this kind.

Gilboa and Zemel (1989) show that finding an equilibrium of a bimatrix game with maximum payoff sum is NP-hard, so for this problem no efficient algorithm is likely to exist. The same holds for other problems that amount essentially to examining all equilibria, like finding an equilibrium with maximum support. For other game-theoretic aspects of computing see Linial (1994) and Koller, Megiddo, and von Stengel (1994).

The usefulness of algorithms for solving games should be tested further in practice. Many of the described methods are being implemented in the project GAMBIT, accessible by internet, and reviewed in McKelvey and McLennan (1996). The GALA system by Koller and Pfeffer (1997) allows to generate large game trees automatically, and solves them according to Theorem 4.4. These program systems are under development to become efficient and easily usable tools for the applied game theorist.

References

V. Aggarwal (1973), On the generation of all equilibrium points for bimatrix games through the Lemke–Howson algorithm. Math. Programming 4, 233–234.

C. Audet, P. Hansen, B. Jaumard, and G. Savard (1996), Enumeration of all extreme equilibrium strategies of bimatrix games. Les Cahiers du GERAD, G-96-32, Montréal.

D. Avis and K. Fukuda (1992), A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. Discrete and Computational Geometry 8, 295–313.

M. Bastian (1976), Another note on bimatrix games. Math. Programming 11, 299–300.

I. M. Bomze (1992), Detecting all evolutionarily stable strategies. Journal of Optimization Theory and Applications 75, 313–329.

A. Charnes (1953), Constrained games and linear programming. Proc. National Academy of Sciences of the U.S.A. 39, 639–641.

V. Chvátal (1983), Linear Programming. Freeman, New York.

R. W. Cottle, J.-S. Pang, and R. E. Stone (1992), The Linear Complementarity Problem. Academic Press, San Diego.

G. B. Dantzig (1963), Linear Programming and Extensions. Princeton University Press, Princeton.

J. Dickhaut and T. Kaplan (1991), A program for finding Nash equilibria. The Mathematica Journal 1:4, 87–93.

B. C. Eaves (1973), Polymatrix games with joint constraints. SIAM J. Appl. Math. 24, 418–423.

I. Gilboa, E. Kalai, and E. Zemel (1990), On the order of eliminating dominated strategies. Operations Research Letters 9, 85–89.

I. Gilboa, E. Kalai, and E. Zemel (1993), The complexity of eliminating dominated strategies. Mathematics of Operations Research 18, 553–565.

I. Gilboa and E. Zemel (1989), Nash and correlated equilibria: some complexity considerations. Games and Economic Behavior 1, 80–93.

J. C. Harsanyi and R. Selten (1988), A General Theory of Equilibrium Selection in Games. MIT Press, Cambridge.

S. Hart (1992), Games in extensive and strategic forms. In: Handbook of Game Theory, Vol. I, eds. R. J. Aumann and S. Hart, Elsevier, Amsterdam, pp. 19–40.

J. Hillas and E. Kohlberg (1998), Foundations of strategic equilibrium. In: Handbook of Game Theory, Vol. III, eds. R. J. Aumann and S. Hart, Elsevier, Amsterdam.

J. T. Howson, Jr. (1972), Equilibria of polymatrix games. Management Science 18, 312–318.

J. T. Howson, Jr., and R. W. Rosenthal (1974), Bayesian equilibria of finite two-person games with incomplete information. Management Science 21, 313–315.

M. J. M. Jansen (1981), Maximal Nash subsets for bimatrix games. Naval Research Logistics Quarterly 28, 147–152.

H. Keiding (1997), On the maximal number of Nash equilibria in a bimatrix game. Games and Economic Behavior, to appear.

D. E. Knuth, C. H. Papadimitriou, and J. N. Tsitsiklis (1988), A note on strategy elimination in bimatrix games. Operations Research Letters 7, 103–107.

E. Kohlberg and J.-F. Mertens (1986), On the strategic stability of equilibria. Econometrica 54, 1003–1037.

M. Kojima, N. Megiddo, T. Noma, and A. Yoshise (1991), A Unified Approach to Interior Point Algorithms for Linear Complementarity Problems. Lecture Notes in Computer Science 538, Springer, Berlin.

D. Koller and N. Megiddo (1992), The complexity of two-person zero-sum games in extensive form. Games and Economic Behavior 4, 528–552.

D. Koller and N. Megiddo (1996), Finding mixed strategies with small supports in extensive form games. International Journal of Game Theory 25, 73–92.

D. Koller, N. Megiddo, and B. von Stengel (1994), Fast algorithms for finding randomized strategies in game trees. Proceedings of the 26th ACM Symposium on Theory of Computing, 750–759.

D. Koller, N. Megiddo, and B. von Stengel (1996), Efficient computation of equilibria for extensive two-person games. Games and Economic Behavior 14, 247–259.

D. Koller and A. Pfeffer (1997), Representations and solutions for game-theoretic problems. Artificial Intelligence 94, 167–215.

I. Krohn, S. Moltzahn, J. Rosenmüller, P. Sudhölter, and H.-M. Wallmeier (1991), Implementing the modified LH algorithm. Applied Math. and Computation 45, 31–72.

H. W. Kuhn (1953), Extensive games and the problem of information. In: Contributions to the Theory of Games II, eds. H. W. Kuhn and A. W. Tucker, Annals of Mathematics Studies 28, Princeton Univ. Press, Princeton, pp. 193–216.

H. W. Kuhn (1961), An algorithm for equilibrium points in bimatrix games. Proc. National Academy of Sciences of the U.S.A. 47, 1657–1662.

C. E. Lemke (1965), Bimatrix equilibrium points and mathematical programming. Management Science 11, 681–689.

C. E. Lemke and J. T. Howson, Jr. (1964), Equilibrium points of bimatrix games. Journal of the Society for Industrial and Applied Mathematics 12, 413–423.

N. Linial (1994), Game-theoretic aspects of computing. In: Handbook of Game Theory, Vol. II, eds. R. J. Aumann and S. Hart, Elsevier, Amsterdam, pp. 1339–1395.

W. F. Lucas (1972), An overview of the mathematical theory of games. Management Science 18, Appendix P, 3–19.

O. L. Mangasarian (1964), Equilibrium points in bimatrix games. Journal of the Society for Industrial and Applied Mathematics 12, 778–780.

O. L. Mangasarian and H. Stone (1964), Two-person nonzero-sum games and quadratic programming. Journal of Mathematical Analysis and Applications 9, 348–355.

R. D. McKelvey and A. McLennan (1996), Computation of equilibria in finite games. In: Handbook of Computational Economics, Vol. I, eds. H. M. Amman, D. A. Kendrick, and J. Rust, Elsevier, Amsterdam, pp. 87–142.

A. McLennan and I.-U. Park (1999), Generic 4×4 two person games have at most 15 Nash equilibria. Games and Economic Behavior 26, 111–130.

P. McMullen (1970), The maximum number of faces of a convex polytope. Mathematika 17, 179–184.

N. Megiddo (1988), A note on the complexity of p-matrix LCP and computing an equilibrium. Research Report RJ 6439, IBM Almaden Research Center, San Jose, California.

N. Megiddo and C. H. Papadimitriou (1989), On total functions, existence theorems and computational complexity (Note). Theoretical Computer Science 81, 317–324.

J.-F. Mertens (1989), Stable equilibria – a reformulation, Part I. Mathematics of Operations Research 14, 575–625.

J.-F. Mertens (1991), Stable equilibria – a reformulation, Part II. Mathematics of Operations Research 16, 694–753.

H. Mills (1960), Equilibrium points in finite games. Journal of the Society for Industrial and Applied Mathematics 8, 397–402.

B. M. Mukhamediev (1978), The solution of bilinear programming problems and finding the equilibrium situations in bimatrix games. U.S.S.R. Computational Mathematics and Mathematical Physics 18, 60–66.

K. Mulmuley (1994), Computational Geometry: An Introduction Through Randomized Algorithms. Prentice-Hall, Englewood Cliffs.

K. G. Murty (1988), Linear Complementarity, Linear and Nonlinear Programming. Heldermann Verlag, Berlin.

J. F. Nash (1951), Non-cooperative games. Annals of Mathematics 54, 286–295.

C. H. Papadimitriou (1994), On the complexity of the parity argument and other inefficient proofs of existence. Journal of Computer and System Sciences 48, 498–532.

T. Parthasarathy and T. E. S. Raghavan (1971), Some Topics in Two-Person Games. American Elsevier, New York.

T. Quint and M. Shubik (1997), A theorem on the number of Nash equilibria in bimatrix game. International Journal of Game Theory 26, 353–359.

T. E. S. Raghavan (1994), Zero-sum two-person games. In: Handbook of Game Theory, Vol. II, eds. R. J. Aumann and S. Hart, Elsevier, Amsterdam, pp. 735–768.

T. E. S. Raghavan (1998), Non-zero-sum two-person games. In: Handbook of Game Theory, Vol. III, eds. R. J. Aumann and S. Hart, Elsevier, Amsterdam.

I. V. Romanovskii (1962), Reduction of a game with complete memory to a matrix game. Soviet Mathematics 3, 678–681.

A. Schrijver (1986), Theory of Linear and Integer Programming. Wiley, Chichester.

R. Selten (1975), Reexamination of the perfectness concept for equilibrium points in extensive games. International Journal of Game Theory 4, 22–55.

R. Selten (1988), Evolutionary stability in extensive two-person games – correction and further development. Mathematical Social Sciences 16, 223–266.

L. S. Shapley (1974), A note on the Lemke–Howson algorithm. Mathematical Programming Study 1: Pivoting and Extensions, 175–189.

L. S. Shapley (1981), On the accessibility of fixed points. In: Game Theory and Mathematical Economics, eds. O. Moeschlin and D. Pallaschke, North-Holland, Amsterdam, pp. 367–377.

M. J. Todd (1976), Comments on a note by Aggarwal. Math. Programming 10, 130–133.

M. J. Todd (1978), Bimatrix games – an addendum. Math. Programming 14, 112–115.

J. A. Tomlin (1978), Robust implementation of Lemke's method for the linear complementarity problem. Mathematical Programming Study 7: Complementarity and Fixed Point Problems, 55–60.

E. van Damme (1987), Stability and Perfection of Nash Equilibria. Springer, Berlin.

E. van Damme (1998), Strategic equilibrium. In: Handbook of Game Theory, Vol. III, eds. R. J. Aumann and S. Hart, Elsevier, Amsterdam.

A. van den Elzen (1993), Adjustment Processes for Exchange Economies and Noncooperative Games. Lecture Notes in Economics and Math. Systems 402, Springer, Berlin.

A. H. van den Elzen and A. J. J. Talman (1991), A procedure for finding Nash equilibria in bi-matrix games. ZOR – Methods and Models of Operations Research 35, 27–43.

A. H. van den Elzen and A. J. J. Talman (1995), An algorithmic approach towards the tracing procedure of Harsanyi and Selten. Discussion paper No. 95111, Center for Economic Research, Tilburg University.

B. von Stengel (1996a), Efficient computation of behavior strategies. Games and Economic Behavior 14, 220–246.

B. von Stengel (1996b), Computing equilibria for two-person games. Technical Report 253, Dept. of Computer Science, ETH Zürich.

B. von Stengel (1999), New maximal numbers of equilibria in bimatrix games. Discrete and Computational Geometry 21, 557–568.

B. von Stengel, A. H. van den Elzen, and A. J. J. Talman (1998), Computing normal form perfect equilibria for extensive two-person games. Dept. of Computer Science, ETH Zürich.

N. N. Vorob'ev (1958), Equilibrium points in bimatrix games. Theory of Probability and its Applications 3, 297–309.

R. Wilson (1972), Computing equilibria of two-person games from the extensive form. Management Science 18, 448–460.

R. Wilson (1992), Computing simply stable equilibria. Econometrica 60, 1039–1070.

H.-M. Winkels (1979), An algorithm to determine all equilibrium points of a bimatrix game. In: Game Theory and Related Topics, eds. O. Moeschlin and D. Pallaschke, North-Holland, Amsterdam, pp. 137–148.

E. B. Yanovskaya (1968), Equilibrium points in polymatrix games (in Russian). Litovskii Matematicheskii Sbornik 8, 381–384 [Math. Reviews 39 #3831].

E. B. Yanovskaya (1970), Quasistrategies in position games. Engineering Cybernetics 1, 11–19.

G. M. Ziegler (1995), Lectures on Polytopes. Graduate Texts in Mathematics, Vol. 152, Springer, New York.