Computing Equilibria for Two-Person Games

Bernhard von Stengel

ETH Zürich

November 25, 1996
(minor corrections added November 11, 1997)

Abstract. This paper is a self-contained survey of algorithms for computing Nash equilibria of two-person games given in normal form or extensive form. The classical Lemke–Howson algorithm for finding one equilibrium of a bimatrix game is presented graph-theoretically as well as algebraically in terms of complementary pivoting. Common definitions of degenerate games are shown as equivalent. Enumeration of all equilibria is presented as a polytope problem. Algorithms for computing simply stable equilibria and perfect equilibria are explained. For games in extensive form, the reduced normal form may be exponentially large. If the players have perfect recall, the sequence form grows linearly with the size of the game tree and can be used instead of the normal form. Theoretical and practical computational issues of these approaches are mentioned.

Author's address:
Institute for Theoretical Computer Science
ETH Zentrum
CH-8092 Zurich
Switzerland

email: stengel@inf.ethz.ch

This paper is the first version of a chapter for the *Handbook of Game Theory*, Vol. 3, eds. R. J. Aumann and S. Hart, North-Holland, Amsterdam, to appear in 1998. A second version (38 pages) is available from the author. This work was supported by a Heisenberg grant from the Deutsche Forschungsgemeinschaft. The author thanks Dick Cottle and Antoon van den Elzen for helpful comments.

Technical Report #253, Departement Informatik, ETH Zürich.
Electronically available from:
ftp://ftp.inf.ethz.ch/pub/publications/tech-reports/2xx

Contents

1. Introduction

2. Normal form games

2.1. Review of linear programming duality
2.2. Payoff maximization and zero-sum games
2.3. Linear complementarity
2.4. The Lemke–Howson algorithm
2.5. Complementary pivoting
2.6. Degenerate games
2.7. Finding all equilibria

3. Stability and equilibrium refinements

3.1. Simply stable equilibria
3.2. Perfect equilibria and the tracing procedure

4. Extensive Games

4.1. Extensive form and reduced normal form
4.2. Sequence form
4.3. Computing equilibria

5. Computational issues

References

1. Introduction

Finding Nash equilibria of normal 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.

In this paper, we survey algorithms for finding Nash equilibria of two-person games. We give an exposition of classical results, in particular of the algorithm by Lemke and Howson (1964) for finding an equilibrium of a bimatrix game, with the goal to make them as accessible as possible. We also present topics and open problems of current interest. We try to *complement* the recent survey by McKelvey and McLennan (1996) on computation of equilibria in games. Therefore, 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.

First, we consider two-person games in normal form. The 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, and then explain the pivoting scheme that represents the algebraic computation. In its basic form, the algorithm requires that the game is *nondegenerate*, meaning that both players always use the same number of pure strategies with positive probability. Nondegeneracy appears in various other forms in the literature. For clarification, we show that most of these notions are equivalent. We also explain the lexicographic method for solving degenerate games. Then we return to a geometric view, namely certain *polytopes* that can be associated with the payoff matrices. The problem of finding *all* equilibria of a bimatrix game can be phrased as an enumeration problem for the vertices of these polytopes, which has been observed by Vorob'ev (1958), Kuhn (1961), and Mangasarian (1964). In this context, Quint and Shubik (1994) have conjectured an upper bound on the number of equilibria of a nondegenerate square bimatrix game, which is still open for some dimensions.

Second, we look at two methods for finding equilibria of normal form games with additional *refinement* properties. A set of equilibria is called *stable* if every game nearby has an equilibrium nearby. This concept, due to Kohlberg and Mertens (1986), is interesting for degenerate games, as they arise, for example, from games in extensive form. Wilson (1992) modified the Lemke–Howson algorithm for computing equilibria that fulfill a weaker notion, called simple stability. These are equilibria that survive certain perturbations of the game which are easily represented by lexicographic methods for degeneracy resolution. Van den Elzen and Talman (1991) presented a complementary pivoting method for finding a *perfect* equilibrium of a bimatrix game. This method also emulates the linear tracing procedure of Harsanyi and Selten (1988) for finding an equilibrium starting from a given *prior* of initially conceived opponent strategies (van den Elzen and Talman, 1995). We discuss only these two papers since the other methods we are aware of use nonlinear methods as for general N-person games.

Third, we consider methods for extensive form games. In principle, such games can be solved by converting them to normal form and then applying the respective algorithms. However, the number of pure strategies is often *exponential* in the size of the extensive game. This holds also for the reduced normal form of an extensive game where pure strategies differing in irrelevant moves are identified. This vast increase in the description of the game can make its solution computationally intractable. It can be circumvented by looking at *sequences* of moves instead of arbitrary combinations of moves as in pure strategies. The realization probabilities of sequences can be characterized by linear equations if the players have perfect recall (Koller and Megiddo, 1992). In turn, the probabilities for sequences define a behavior strategy for each player. This defines the *sequence form* of the game that is analogous to the normal form but has small size. The solution of games in normal form (in the way we present it here, using linear programming duality) can analogously be applied to the sequence form. The equilibria of a zero-sum game are the solutions to a linear program that has the same size as the extensive game (Romanovskii, 1962; von Stengel, 1996). The complementary pivoting algorithm by Lemke (1965) is applied by von Stengel, van den Elzen and Talman (1996) to the sequence form of non-zero-sum games, analogous to the algorithm by van den Elzen and Talman (1991) presented in Section 3.2.

For two-person games, finding equilibria amounts to finding the pure strategies (or sequences) that are played with positive probability, and solving corresponding linear equations. We try to present this approach in a unified manner, citing and explaining the pertinent papers. We will mention very briefly other works, but refer to McKelvey and McLennan (1996) for further references. In the last section, we will touch the issue of computational complexity, and mention ongoing implementations of the theoretically known methods.

2. Normal form games

In the following sections, we develop the problem of equilibrium computation using the duality theory of linear programming, which we briefly review. We start with the linear program (LP), with mixed strategy probabilities and payoffs as variables, of maximizing the payoff of a single player against a *fixed* opponent strategy. It is then easy to demonstrate that the equilibria of a zero-sum game are the solutions to an LP. The equilibria of non-zero-sum game are the solutions to a linear complementarity problem (LCP). In this context, the LCP specifies that each pure strategy is either played with probability zero or has maximum payoff. Lemke and Howson (1964) gave an algorithm for finding one solution to this LCP. Following Shapley (1974), we explain this algorithm geometrically with labels marking best response regions. The numerical computations are iterative changes, called *pivoting*, to solutions of linear equations. This algebraic implementation is shown next. The uniqueness of the computation requires that the game is *nondegenerate*, which is true for "almost all" games. We discuss the interrelationships of various definitions of nondegeneracy used in the literature. Degenerate games can be solved by a lexicographic method that simulates a perturbation of the equations and is best understood in algebraic terms. The problem of finding *all* equilibria of a bimatrix game can be phrased as an enumeration problem for the vertices of a polytope. We present this geometric view and related results and conjectures on the maximal number of equilibria of a nondegenerate bimatrix game.

2.1. Review of linear programming duality

Our notation for two-person games employs vectors and matrices. All vectors are column vectors, so an n-vector x is treated as an $n \times 1$ matrix. Transposition gives the corresponding row vector $x^\top$. A vector or matrix with all components zero is denoted $\mathbf{0}$, with varying dimension depending on the context. Inequalities like $x \geq \mathbf{0}$ hold componentwise.

A *linear program* (LP) is given by a matrix A and vectors b and c of suitable dimension and states

$$\begin{aligned} \text{maximize} \quad & c^\top x \\ \text{subject to} \quad & Ax = b\,, \\ & x \geq \mathbf{0}. \end{aligned} \tag{2.1}$$

A vector x fulfilling the constraints $Ax = b$ and $x \geq \mathbf{0}$ is called *feasible* for this LP. The *dual* linear program is motivated by finding an upper bound, if it exists, for the objective function $c^\top x$ in this optimization problem. Let A have m rows, and let $y \in \mathbb{R}^m$. Then any feasible x fulfills $y^\top Ax = y^\top b$ and $(y^\top A)x \geq c^\top x$ provided $y^\top A \geq c^\top$. Thus, the objective function $c^\top x$ has the upper bound $y^\top b$ which is minimized in the following *dual LP* for (2.1):

$$\begin{aligned} \text{minimize} \quad & y^\top b \\ \text{subject to} \quad & y^\top A \geq c^\top. \end{aligned} \tag{2.2}$$

Problem (2.1) is also called the *primal* LP. The fact that, for feasible solutions, primal and dual objective function values are mutual bounds is called *weak duality*.

Theorem 2.1. *(Weak duality theorem of Linear Programming.) Consider a feasible solution x to the primal LP (2.1) and a feasible solution y to the dual LP (2.2). Then $c^\top x \leq y^\top b$.*

The central property of linear programs is *strong duality*, which says that the optimal values of primal and dual LP, if they exist, coincide. For a proof see, for example, Schrijver (1986).

Theorem 2.2. *(Strong duality theorem of Linear Programming.) If the primal LP (2.1) and the dual LP (2.2) both have feasible solutions, then they have the same optimal value of their objective functions. That is, there is a feasible primal-dual pair x, y with $c^\top x = y^\top b$.*

The primal LP (2.1) describes the maximization of a linear function of non-negative variables subject to linear equalities, whereas the dual LP (2.2) describes the minimization of a linear function of unconstrained variables subject to linear inequalities. The LP (2.1) is said to be in *equality* form (nonnegativity of x being considered a separate property), and (2.2) is called an LP in *inequality* form. These differences are not essential. Changing signs turns maximization into minimization and reverses inequalities. (Also, the constraints in (2.2) are transposed, which we will do as it is convenient.) A *symmetric* primal-dual pair of linear programs is obtained from the following primal LP in inequality form with nonnegative variables: maximize $c^\top x$ subject to $Ax \le b$, $x \ge \mathbf{0}$. It is converted to equality form by introducing an m-vector z of *slack* variables (assuming A has m rows), so that $Ax \le b$ is equivalent to $Ax + Iz = b$, $z \ge \mathbf{0}$, with the $m \times m$ identity matrix I. It is easy to see that the dual LP is then equivalent to the LP: minimize $y^\top b$ subject to $y^\top A \ge c^\top$, $y \ge \mathbf{0}$. Furthermore, the dual of the dual is again the primal. In general, an LP may involve both nonnegative and unconstrained variables subject to linear equalities as well as inequalities, and is then called a *mixed* LP. In a primal-dual pair of mixed LPs, unconstrained dual variables correspond to primal equalities and nonnegative dual variables correspond to primal inequalities and vice versa (see Figure 2.1 below for an example). Weak and strong duality continue to hold and are easily derived from the theorems above.

2.2. Payoff maximization and zero-sum games

Let (A, B) be a two-person game in normal form (also called a bimatrix game), where A is an $m \times n$ matrix of payoffs for player 1 and B is an $m \times n$ matrix of payoffs for player 2. The m rows are the pure strategies of player 1 and the n columns are the pure strategies of player 2. A mixed strategy x for player 1 can be represented by the vector x of probabilities for playing pure strategies. Thus, x is an m-vector fulfilling $x \ge \mathbf{0}$ and $\mathbf{1}_m^\top x = 1$ where $\mathbf{1}_m$ is an m-vector with all components equal to one. If there is no ambiguity about the dimension, we write $\mathbf{1}$ instead of $\mathbf{1}_m$. Similarly, a mixed strategy y for player 2 can be represented as an n-vector y fulfilling $y \ge \mathbf{0}$ and $\mathbf{1}_n^\top y = 1$.

When player 1 and player 2 use the mixed strategies x and y, their expected payoffs are $x^\top Ay$ and $x^\top By$, respectively. Strategy x is a *best response* to y if it maximizes the expression $x^\top(Ay)$, for fixed y. Similarly, a best response y of player 2 to x maximizes $(x^\top B)y$. An *equilibrium* is a pair (x, y) of mutual best responses.

If the strategy y of player 2 is known, then a best response x of player 1 to y is a solution to the following LP:

$$\begin{aligned} \text{maximize} \quad & x^\top(Ay) \\ \text{subject to} \quad & x^\top \mathbf{1} = 1, \\ & x \quad \geq \mathbf{0}. \end{aligned} \tag{2.3}$$

We consider the dual of this LP, which has only a single dual variable u:

$$\begin{aligned} \text{minimize} \quad & u \\ \text{subject to} \quad & \mathbf{1}u \geq Ay. \end{aligned} \tag{2.4}$$

Both LPs are feasible. By strong duality (Theorem 2.2), they have the same optimal value. That is, the maximal expected payoff $x^\top Ay$ to player 1 is the same as the smallest u fulfilling $\mathbf{1}u \geq Ay$. The latter is obviously the maximum of the entries of the vector Ay, which are the expected payoffs to player 1 for his pure strategies.

Consider now a *zero-sum game*, where $B = -A$. Player 2, when choosing y, has to assume that his opponent plays rationally and maximizes $x^\top Ay$. This maximum payoff to player 1 is the optimal value of the LP (2.3), which is equal to the optimal value u of the dual LP (2.4). Player 2 is interested in minimizing u by his choice of y. The constraints of (2.4) are linear in u and y even if y is treated as a variable, which has to represent a mixed strategy. So a minmax strategy y of player 2 (minimizing the maximum amount he has to pay) is a solution to the mixed LP with variables u, y

$$\begin{aligned} \text{minimize} \quad & u \\ \text{subject to} \quad & \mathbf{1}_n^\top y = 1, \\ & \mathbf{1}_m u - Ay \geq \mathbf{0}, \\ & y \geq \mathbf{0}. \end{aligned} \tag{2.5}$$

Figure 2.1 shows on the left a simple example of the LP (2.5) for a 3×2 game where the matrix of payoffs to player 1 is

$$A = \begin{bmatrix} 0 & 6 \\ 1 & 4 \\ 3 & 3 \end{bmatrix}.$$

The LP variables u, y_1, y_2 and the sign restrictions $y \geq \mathbf{0}$ are marked at the top. The coefficients of the objective function are written at the bottom, separated by a line. The right part of Figure 2.1 shows a similar diagram to be read vertically. It

$$
\begin{array}{c|cc|c|c}
 & \geq 0 & \geq 0 & & \\
u & y_1 & y_2 & & \\
\hline
0 & 1 & 1 & = & 1 \\
\hline
1 & 0 & -6 & & 0 \\
1 & -1 & -4 & \geq & 0 \\
1 & -3 & -3 & & 0 \\
\hline
\hline
1 & 0 & 0 & \to & \min
\end{array}
\qquad\qquad
\begin{array}{cc|c|cc|c}
 & v & 0 & 1 & 1 & 1 \\
\hline
\geq 0 & x_1 & 1 & 0 & -6 & 0 \\
\geq 0 & x_2 & 1 & -1 & -4 & 0 \\
\geq 0 & x_3 & 1 & -3 & -3 & 0 \\
\hline
 & & \shortparallel & \multicolumn{2}{c|}{\wedge\!\mid} & \downarrow \\
 & & 1 & 0 & 0 & \max
\end{array}
$$

Figure 2.1. Example of the LP (2.5) for a 3×2 zero-sum game, and its dual LP.

represents the dual of the LP (2.5) with variables v (a scalar corresponding to the equation $\mathbf{1}_n^\top y = 1$) and x, which has the form

$$
\begin{array}{ll}
\text{maximize} & v \\
\text{subject to} & \qquad\quad x^\top \mathbf{1}_m = 1, \\
 & v\mathbf{1}_n^\top - x^\top A \ \leq \mathbf{0}, \\
 & \qquad\quad x \qquad \geq \mathbf{0}.
\end{array}
\tag{2.6}
$$

It is easy to verify that this LP describes the problem of finding a maxmin strategy x (with maxmin payoff 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) iff (if and only if) u, y is an optimal solution of the LP (2.5) and v, x is an optimal solution of its dual LP (2.6). Thereby, u is the maxmin payoff to player 1, v is the minmax payoff to player 2, and $u = v$, denoting the value of the game.*

Thus, the "maxmin = minmax" theorem for zero-sum games follows directly from LP duality. This connection was noted by von Neumann and Dantzig in the late forties when linear programming took its shape (see Dantzig, 1991, p. 24). Conversely, linear programs can be expressed as zero-sum games (Gale, Kuhn, and Tucker, 1950).

There are standard algorithms for solving LPs, in particular the Simplex algorithm (see Dantzig, 1963). Usually, they compute a primal solution together with a dual solution since this proves that the optimum is reached.

2.3. Linear complementarity

A best response x of player 1 against the mixed strategy y of player 2 is a solution to the LP (2.3). This is also useful for games that are not zero-sum. By strong duality, a feasible solution x is optimal iff there is a dual solution u fulfilling $\mathbf{1}_m u \geq Ay$ and $x^\top(Ay) = u$, that is, $x^\top(Ay) = (x^\top \mathbf{1}_m)u$ or equivalently

$$x^\top(\mathbf{1}_m u - Ay) = 0\,. \tag{2.7}$$

This condition states that x and $\mathbf{1}_m u - Ay$ are orthogonal. Because these vectors are nonnegative, 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. Here, we know that x, being a mixed strategy, has at least one positive component, so the respective component of $\mathbf{1}_m u - Ay$ is zero and u is the maximum of the entries of Ay. Any pure strategy i, $1 \leq i \leq m$, of player 1 is a best response to y iff the ith component of the slack vector $\mathbf{1}_m u - Ay$ is zero. So (2.7) amounts to the following well-known property (Nash, 1951): A strategy x is a best response to y iff it only plays pure strategies that are best responses with positive probability.

For player 2, strategy y is a best response to x iff it maximizes $(x^\top B)y$ subject to $\mathbf{1}_n^\top y = 1$, $y \geq \mathbf{0}$. This gives an LP analogous to (2.3). Its dual LP has a single scalar v as variable and says: minimize v subject to $\mathbf{1}_n v \geq B^\top x$. A primal-dual pair y, v of feasible solutions is optimal iff, analogous to (2.7),

$$y^\top(\mathbf{1}_n v - B^\top x) = 0\,. \tag{2.8}$$

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

Theorem 2.4. *The vector pair (x, y) is an equilibrium of the bimatrix game (A, B) iff there are reals u, v such that*

$$\begin{array}{rrrl}
 & \mathbf{1}_m^\top x & & = 1 \\
 & & \mathbf{1}_n^\top y & = 1 \\
\mathbf{1}_m u & & -\,Ay & \geq \mathbf{0} \\
 \mathbf{1}_n v & -\,B^\top x & & \geq \mathbf{0} \\
 & x\,, & y & \geq \mathbf{0}
\end{array} \tag{2.9}$$

and (2.7), (2.8) hold.

The constraints (2.9) are linear in the variables u, v, x, y. The orthogonality conditions (2.7) and (2.8) state that the nonnegative vector of slacks for these constraints, $(\mathbf{1}_m^\top x - 1, \mathbf{1}_n^\top y - 1, \mathbf{1}_m u - Ay, \mathbf{1}_n v - B^\top x)$, is complementary to the vector

(u, v, x, y) of variables. Such a problem is called a *linear complementarity problem* (LCP). The complementarity condition is nonlinear in the variables. However, it is comparatively simple since it amounts to the combinatorial problem of deciding which pure strategies may have positive probability. 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.

2.4. The Lemke–Howson algorithm

In their seminal paper, Lemke and Howson (1964) described an algorithm for finding one equilibrium of a bimatrix game. Shapley (1974) gave an intuitive explanation of this algorithm which is easily visualized for games of small dimension. We will follow his exposition here. For illustration, we use a simple example of a 3×2 bimatrix game (A, B) with

$$A = \begin{bmatrix} 0 & 6 \\ 2 & 5 \\ 3 & 3 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 4 & 3 \end{bmatrix}. \tag{2.10}$$

Throughout, we will use the following notation. The given bimatrix game is (A, B) and has size $m \times n$. The sets of pure strategies are $I = \{1, \ldots, m\}$ for player 1 and $N = \{1, \ldots, n\}$ for player 2. The rows of A are denoted a_i for $i \in I$ and the columns of B are denoted B_j for $j \in N$. The sets of mixed strategies for player 1 and 2 are denoted by

$$X = \{x \in \mathbb{R}^m \mid \mathbf{1}_m^\top x = 1,\ x \geq \mathbf{0}\}, \qquad Y = \{y \in \mathbb{R}^n \mid \mathbf{1}_n^\top y = 1,\ y \geq \mathbf{0}\}.$$

For easier distinction of the pure strategies of the players, let $J = \{m+1, \ldots, m+n\}$ be a "copy" of N where any j in N is identified with $m + j$ in J. A *label* is any element of $I \cup J$. In our example, labels $1, 2, 3$ refer to the pure strategies of player 1 and $4, 5$ to those of player 2.

The labels are used to mark the points in X and Y, as follows. Consider Y, which is in our example the line segment in $\mathbb{R}^2$ connecting $(1, 0)^\top$ and $(0, 1)^\top$. We divide Y into regions $Y(i)$ for $i \in I$ where the pure strategy i of player 1 is a best response, so for $i \in I$,

$$Y(i) = \{y \in Y \mid a_i y \geq a_k y \ \text{ for all } k \in I\}.$$

Each best response region $Y(i)$ is therefore a polytope (a bounded set defined by linear inequalities), and Y is the union of these regions. In our example,

$$\begin{aligned} Y(1) &= \{(y_1, y_2)^\top \in Y \mid 0 \leq y_1 \leq 1/3\} \\ Y(2) &= \{(y_1, y_2)^\top \in Y \mid 1/3 \leq y_1 \leq 2/3\} \\ Y(3) &= \{(y_1, y_2)^\top \in Y \mid 2/3 \leq y_1 \leq 1\}. \end{aligned}$$

These sets are one-dimensional. In general, the set $Y(i)$ for $i \in I$ is either empty or $(n-1)$-dimensional, except in degenerate cases. (The *dimension* of convex sets like these is defined as follows: A set has dimension d iff it has $d+1$, but no more, affinely independent points. Affine independence means no point is an affine combination of others. An affine combination of points $x^1, \ldots, x^k$ is given by $\sum_{i=1}^k x^i \lambda_i$ where $\lambda_1, \ldots, \lambda_k$ are arbitrary reals with $\sum_{i=1}^k \lambda_i = 1$.)

Secondly, we consider sets $Y(j)$ for the strategies $j \in J$ of player 2 himself containing those y where j has probability zero. In order to obtain the same dimension for these sets as for the sets $Y(i)$, it is not required that y belongs to Y. For $j \in J$, let

$$Y(j) = \{y = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n \mid \mathbf{1}_n^\top y \le 1,\ y \ge \mathbf{0},\ y_{j-m} = 0\,\}.$$

We say y has label k if $y \in Y(k)$, for $k \in I \cup J$, and define the set of labels of y as

$$L(y) = \{\, k \in I \cup J \mid y \in Y(k)\,\}. \tag{2.11}$$

Figure 2.2 shows on the right Y as a subset of $\mathbb{R}^n$ for our example (with $n = 2$) where the sets $Y(k)$ are indicated by their labels k which are drawn as circled numbers. Some points y have more than one label, for example $y = (1,0)^\top$ has the set of labels $L(y) = \{3, 5\}$.

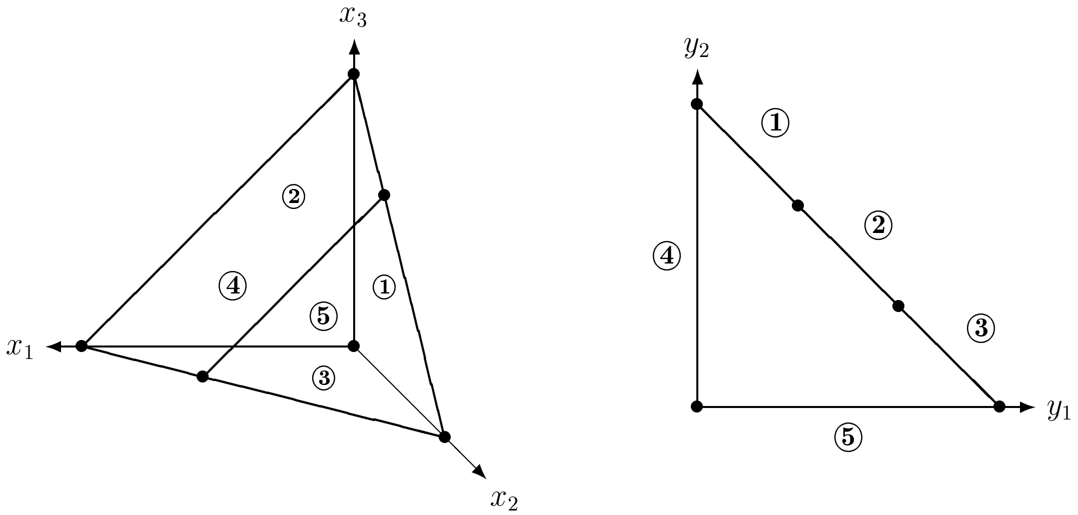

Figure 2.2. Strategy spaces X and Y of the players for the bimatrix game (A, B) in (2.10). The labels $1, 2, 3$ (drawn as circled numbers) are the pure strategies of player 1 and marked in X where they have probability zero, in Y where they are best responses. The pure strategies of player 2 are similar labels $4, 5$. The dots mark points x and y with a maximum number of labels.

The strategy set X is divided analogously into regions $X(j)$ where each pure strategy j of player 2 is a best response. Then for $j \in J$,

$$X(j) = \{x \in X \mid x^\top B_{j-m} \ge x^\top B_{k-m} \text{ for all } k \in J\}.$$

Similarly, for $i \in I$,

$$X(i) = \{x = (x_1, \ldots, x_m)^\top \in \mathbb{R}^m \mid \mathbf{1}_m^\top x \le 1,\ x \ge \mathbf{0},\ x_i = 0\}\,.$$

We say x has label k if $x \in X(k)$, for $k \in I \cup J$. The set of labels of x is denoted

$$L(x) = \{\, k \in I \cup J \mid x \in X(k)\,\}, \tag{2.12}$$

where it will always be clear from the context that $L(y)$ is defined by (2.11) and $L(x)$ by (2.12). In our example,

$$X(4) = \{(x_1, x_2, x_3)^\top \in X \mid x_1 + 4x_3 \ge 2x_2 + 3x_3\},$$

and $X(5)$ is the complement of $X(4)$ in X together with those points where both strategies of player 2 are best responses; the latter are the points in $X(4) \cap X(5)$ which is the line segment connecting $({}^2\!/_3, {}^1\!/_3, 0)^\top$ and $(0, {}^1\!/_3, {}^2\!/_3)^\top$.

Figure 2.2 shows on the left X as a subset of $\mathbb{R}^3$. The labels 4 and 5 mark the subsets of the triangle X that are the best response regions of player 2. The labels $1, 2, 3$ mark the sets $X(1)$, $X(2)$, $X(3)$, which are the right, back, and bottom sides of the pyramid $\{x \in \mathbb{R}^3 \mid \mathbf{1}^\top x \le 1,\ x \ge \mathbf{0}\}$.

With the help of these labels, it is easy to identify the equilibria of the game. These are exactly the pairs (x, y) in $X \times Y$ that are *completely labeled*, that is, $L(x) \cup L(y) = I \cup J$. The reason is that in equilibrium, a pure strategy is either a best response or has probability zero (or both, which can only happen if the game is degenerate, as we will explain). A *missing* label k represents a pure strategy (of either player) that has positive probability but is not a best response, which is forbidden in equilibrium.

There are not many points x and y such that (x, y) can be an equilibrium. Together, they must have $m + n$ labels, meaning they belong to several sets $X(k)$ and $Y(l)$, respectively. For any nonempty set K of labels, $\emptyset \ne K \subseteq I \cup J$, let

$$X(K) = \bigcap_{k \in K} X(k), \qquad Y(K) = \bigcap_{k \in K} Y(k).$$

In our example in Figure 2.2, any two of the sets $X(k)$ intersect, if at all, in a line segment, any three of them in a point. Similarly, any two sets $Y(k)$ intersect at most in a point.

In general, the dimension of $X(K)$ and $Y(K)$ should decrease with the size of K. This is usually, but not always the case. We have to assume that the game is *nondegenerate*.

Definition 2.5. *The $m \times n$ bimatrix game (A, B) is called nondegenerate if it has the following properties: whenever $\emptyset \ne K \subseteq I \cup J$ and $K = L(x)$ for some $x \in \mathbb{R}^m$,* *then $X(K)$ has dimension $m-|K|$, and if $K=L(y)$ for some $y \in \mathbb{R}^n$, then $Y(K)$ has dimension $n-|K|$.*

Until further notice, we assume that the game is nondegenerate. Then $|L(x)| \le m$ for every x in X, and $|L(x)| = m$ holds for finitely many points x in X, given by the nonempty sets $\{x\} = X(K)$ with $|K| = m$, $K \subseteq I \cup J$. Similarly, $|L(y)| \le n$ for all y in Y, and the nonempty sets $Y(K)$ where $|K| = n$ identify those points in Y that have exactly n labels. In Figure 2.2, inspection shows the following completely labeled pairs in $X \times Y$: $(x^1, y^1) = \big((0,0,1)^\top, (1,0)^\top\big)$ where $L(x^1) = \{1,2,4\}$, $L(y^1) = \{3,5\}$, furthermore $(x^2, y^2) = \big((0, 1/3, 2/3)^\top, (2/3, 1/3)^\top\big)$ where $L(x^2) = \{1,4,5\}$, $L(y^2) = \{2,3\}$, and $(x^3, y^3) = \big((2/3, 1/3, 0)^\top, (1/3, 2/3)^\top\big)$ where $L(x^3) = \{3,4,5\}$, $L(y^3) = \{1,2\}$. This geometric-qualitative inspection is very suitable for studying equilibria of games of size up to 3×3.

The Lemke–Howson method is not such an inspection of candidate points but a search along a piecewise linear path, as follows. Fix an arbitrary label $k \in I \cup J$, and let

$$M(k) = \{\, (x,y) \mid I \cup J - \{k\} \subseteq L(x) \cup L(y) \,\}. \tag{2.13}$$

That is, $M(k)$ consists of *almost completely labeled* pairs (x,y) that have all labels except possibly k, which is called the *missing* label. The set $M(k)$ includes all equilibria. Furthermore, $M(k)$ contains pairs (x,y) where x is not in X and y is not in Y, in particular $(x,y) = (\mathbf{0},\mathbf{0})$ which is the element of $X(I) \times Y(J)$. We call $(\mathbf{0},\mathbf{0})$ the *artificial equilibrium* since it is a completely labeled pair. Any (x,y) in $M(k)$ has at least $m+n-1$ labels. By nondegeneracy, this means either $m-1$ or more labels for x and n labels for y, or m labels for x and at least $n-1$ labels for y. In either case, the missing label gives a degree of freedom to change x or y, respectively, which makes it possible to follow a path defined by $M(k)$.

We explain this first for our example with the help of Figure 2.3. The algorithm starts from $(\mathbf{0},\mathbf{0})$. Let 2 be the label that may be dropped. This means x_2 can be increased, which we do until $x_2 = 1$, reaching the vertex $x = (0,1,0)^\top$ of X. This is shown as step I in Figure 2.3, where $y = (0,0)^\top$ stays fixed. For this pair (x,y) that is reached, x has m labels $1,3,5$ and y has n labels $4,5$. Since label 2 is missing, one of the $m+n$ labels is duplicate, which is label 5 picked up in X by moving x from $\mathbf{0}$ to $(0,1,0)^\top$. This duplicate label is now dropped from y which is moved from $(0,0)^\top$ to $(0,1)^\top$ while x stays fixed (step II). Thereby, $L(2)$ is reached as a subset of $X \times Y$. (Equivalently, one may start directly with the pure strategy $(0,1,0)^\top$ in X representing the missing label 2 and its best response $(0,1)^\top$ in Y.) Now, label 1 is duplicate since it was just picked up by y, so it may be dropped from x while keeping the remaining labels $3,5$ (step III). Thereby, the probability x_1 (for the newly found best response strategy 1 to y) is increased until $x = x^3$, picking up another label, 4, which is now duplicate. This means 4 can be dropped from y which

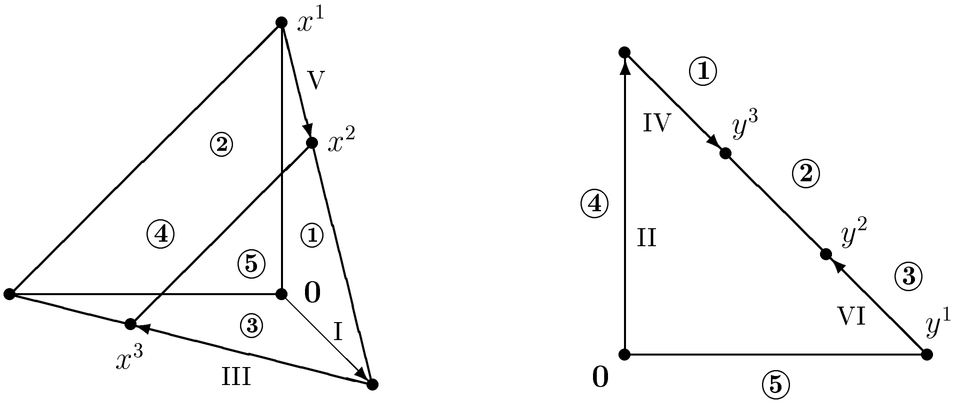

Figure 2.3. The set $M(2)$ of almost completely labeled pairs where label 2 is missing is formed by the paths with line segments (in the product graph) I–II–III–IV, connecting the artificial equilibrium $(\mathbf{0},\mathbf{0})$ and (x^3,y^3), and V–VI, connecting equilibria (x^1,y^1) and (x^2,y^2).

is moved from $(0,1)^\top$ to y^3. The label encountered there is the missing label 2, so the path terminates at the completely labeled pair (x^3,y^3) which is the equilibrium found.

The Lemke–Howson algorithm, although we described it by "moving" along edges, is combinatorial and can it its general form be described in graph-theoretic terms. Let G_1 be the graph whose vertices are those points x with $|L(x)| = m$. Except for $\mathbf{0}$ in $X(I)$, these points belong to X. Any two such vertices x, x' are joined by an edge if they differ in exactly one label. The set $K = L(x) \cap L(x')$ of the $m-1$ labels that x and x' have in common define the line segment $X(K)$ which represents this edge in $\mathbb{R}^m$. The vertices x and x' are the endpoints of this edge (and not interior points) since it is not possible to get additional labels by taking convex combinations. Similarly, let G_2 be the graph with vertices y where $|L(y)| = n$, and edges joining those vertices that have $n-1$ common labels. In our example, Figure 2.2 (or 2.3) is a picture of these graphs G_1 and G_2. Finally, we consider the *product graph* G of G_1 and G_2. Its vertices are (x,y) where x is a vertex of G, and y is a vertex of G_2. The edges of G are given by $\{x\} \times e_2$ for vertices x of G_1 and edges e_2 of G_2, or $e_1 \times \{y\}$ for edges e_1 of G_1 and vertices y of G_2.

Theorem 2.6. *(Lemke and Howson, 1964; Shapley, 1974.) Let (A,B) be a nondegenerate bimatrix game and k be a label in $I \cup J$. Then $M(k)$ in (2.13) consists of disjoint paths and loops in the product graph G of G_1 and G_2. The endpoints of the paths are the equilibria of the game and the artificial equilibrium $(\mathbf{0},\mathbf{0})$. The number of equilibria of the game is odd.*

Proof. Let $(x,y) \in M(k)$. Then x and y have together $m+n$ or $m+n-1$ labels.

a) If the pair (x, y) is completely labeled, it is a vertex of G and either an equilibrium or $(\mathbf{0}, \mathbf{0})$.

Otherwise, $L(x) \cup L(y) = I \cup J - \{k\}$ and there are the following possibilities:

b) $|L(x)| = m$, so x is a vertex of G_1, and y has $n - 1$ labels, which define an edge e_2 of G_2, so that $\{x\} \times e_2$ is an edge of G, or

c) x has $m - 1$ labels and is part of an edge e_1 of G_1 and y has n labels and is a vertex of G_2, so that $e_1 \times \{y\}$ is an edge of G, or

d) $|L(x)| = m$ and $|L(y)| = n$, so that (x, y) is a vertex of G.

Thus, $M(k)$ defines a subgraph of G. In case a), the vertex (x, y) is incident to a unique edge in the subgraph $M(k)$, namely $\{x\} \times Y(L(y) - \{k\})$ if $k \in L(y)$ or $X(L(x) - \{k\}) \times \{y\}$ if $k \in L(x)$, respectively. In case d), $L(x) \cup L(y) = I \cup J - \{k\}$ so there is a duplicate label l in $L(x) \cap L(y)$. Then (x, y) is incident to the two edges $\{x\} \times Y(L(y) - \{l\})$ and $X(L(x) - \{l\}) \times \{y\}$ in $M(k)$. So $M(k)$ is a graph where all vertices are incident to one or two edges, so $M(k)$ consists of paths and loops. The endpoints of the paths are the equilibria and the artificial equilibrium $(\mathbf{0}, \mathbf{0})$. Clearly, the number of these endpoints is even, so the number of equilibria is odd. □

This theorem provides a constructive, elementary proof that every nondegenerate game has an equilibrium, independently of the result of Nash (1951). Algorithmically, the subgraph $M(k)$ of G in Theorem 2.6 is generated implicitly as the computation goes along. (We show its algebraic aspects in the next section.) Starting from a completely labeled vertex of G, the edges in G of the path that is followed are alternatingly edges of G_1 and G_2, leaving the vertex in the other graph fixed. The first edge is defined by the label k that is dropped initially, the subsequent edges are defined by the duplicate labels encountered along the way.

In Figure 2.3, we have shown the path that starts from $(\mathbf{0}, \mathbf{0})$ with steps I–II–III–IV. Starting from the equilibrium (x^1, y^1), $M(2)$ leads by steps V and VI to the equilibrium (x^2, y^2). This can also be used algorithmically, since (x^1, y^1) is reached from $(\mathbf{0}, \mathbf{0})$ by dropping another label k, for example 3 (or 1 or 4) instead of 2. However, it can be shown that this does not necessarily generate all equilibria, that is, the union of the paths in the subgraphs $M(k)$ for all $k \in I \cup J$ may be disconnected (see Aggarwal, 1973). Variants of the Lemke–Howson method have similar properties (Bastian, 1976; Todd, 1976, 1978). Shapley (1981) discusses more general methods as a potential way to overcome this problem.

Shapley (1974) introduced the *index* of an equilibrium of a nondegenerate bimatrix game as the index of a certain determinant defined by the equilibrium and the payoff matrices. The index does not depend on the order of pure strategies. For any missing label k, the two ends of an almost completely labeled path in $M(k)$ (as in Theorem 2.6) have opposite index. The artificial equilibrium has positive index (by Shapley's definition; sometimes the opposite sign is used). Thus, the Lemke–Howson algorithm started from $(\mathbf{0},\mathbf{0})$ computes always an equilibrium with negative index. For any equilibrium with negative index that is found, the missing label used in another computation can be used to find a corresponding equilibrium of positive index, and vice versa. Thus, using the Lemke–Howson method iteratively with all labels k in $I \cup J$, and from all equilibria that are found as starting points, finds an odd number of equilibria, one more with negative index than with positive index.

2.5. Complementary pivoting

The Lemke–Howson algorithm is implemented by iteratively computing *basic solutions* to a system of linear equations equivalent to (2.9). These basic solutions represent the graph vertices used above, and moving along a graph edge is a basis change known as *pivoting*. We present this algebraic version of the algorithm since it is necessary for the computation, and since it is readily extended to degenerate games.

Pivoting is well known as the main step of the Simplex algorithm for solving linear programs given in equality form. The linear constraints (2.9) are also converted to equality form by introducing vectors of slack variables: If there is z in $\mathbb{R}^n$ with

$$\begin{aligned} \mathbf{1}_m^\top x \qquad\quad &= 1 \\ -\mathbf{1}_n v + B^\top x + I_n z &= \mathbf{0} \\ x, \qquad z &\geq \mathbf{0} \end{aligned} \tag{2.14}$$

(where I_n is the $n \times n$ identity matrix), and w in $\mathbb{R}^m$ with

$$\begin{aligned} \mathbf{1}_n^\top y \qquad\quad &= 1 \\ -\mathbf{1}_m u + Ay + I_m w &= \mathbf{0} \\ y, \qquad w &\geq \mathbf{0} \end{aligned} \tag{2.15}$$

then u, v, x, y is a solution to (2.9), and vice versa.

Both (2.14) and (2.15), and the two systems taken together, are of the form $Dr = b$ with a matrix D, a vector r of variables r_j, some of which are nonnegative, and a right hand side b. The matrix D has full rank, so that b belongs always to the space spanned by the columns D_j of D. A *basis* β is given by a basis $\{D_j \mid j \in \beta\}$ of this column space, so that the square matrix D_β formed by these columns is invertible. The corresponding *basic solution* is the unique vector $r_\beta = (r_j)_{j\in\beta}$ with $D_\beta r_\beta = b$, where the variables r_j for j in β are called *basic variables*, and $r_j = 0$ for all *nonbasic* variables r_j, $j \notin \beta$, so that $Dr = b$. If this solution fulfills also the nonnegativity constraints, then the basis β is called *feasible*. If β is a basis for the system $Dr = b$, then the corresponding basic solution can be read from the equivalent system $D_\beta^{-1} Dr = D_\beta^{-1} b$ since the columns of $D_\beta^{-1} D$ (for the basic variables) form the identity matrix. This new system, called a *tableau*, can also be written as

$$r_\beta = D_\beta^{-1} b - \sum_{j \notin \beta} D_\beta^{-1} D_j r_j. \tag{2.16}$$

We require that an *unconstrained* variable, like v in (2.14), is always basic. In a basic solution to the $n+1$ equations in (2.14), $n+1$ of the $m+n+1$ variables are basic, including v and at least one variable x_i since $x \neq \mathbf{0}$. Thus, at least one of the n slack variables z_j is nonbasic and zero, so that in a feasible solution v is minimal subject to $\mathbf{1}_n v \geq B^\top x$ and represents the best response payoff to player 2 against x as we desire. To avoid special treatment of the variable v, one can make the payoff matrix positive by adding a constant to all entries (which does not change the game), so that $B > \mathbf{0}$ and v is positive and therefore basic in any basic feasible solution. Similarly, assuming $A > \mathbf{0}$ implies that u in (2.15) is always positive.

In a basic feasible solution v, x, z to (2.14), the m nonbasic variables are zero and represent the *labels* of x, namely the pure strategies i of player 1 that have probability zero (if $x_i = 0$), or pure best responses j of player 2 to x (if $z_j = 0$). In a nondegenerate game, m is the maximum number of labels of x, so that all basic variables are positive (except possibly v where the sign does not matter). Similarly, basic feasible solutions to (2.15) represent maximally labeled mixed strategies y of player 2.

In terms of basic solutions, the Lemke–Howson algorithm for the example (2.10) illustrated in Figure 2.3 works as follows. Since x and y are mixed strategies in any solution to (2.14) and (2.15), we do not use the artificial equilibrium but compute separately the first two steps such that $x \in X$ and $y \in Y$. That is, the initial solutions to (2.14) and (2.15), determined by steps I and II, are given by the pure strategy $x = (0,1,0)^\top$ resulting from the missing label, and its best response $y = (0,1)^\top$. The corresponding basic variables in (2.14) are v, x_2, z_1. System (2.14), besides the conditions $x, z \geq \mathbf{0}$, is here

$$\begin{aligned}
x_1 + x_2 + x_3 \quad &= 1 \\
-v + x_1 + 4x_3 + z_1 \quad &= 0 \\
-v + 2x_2 + 3x_3 + z_2 &= 0
\end{aligned}$$

For the initial basis, it has the equivalent form (2.16) given by the tableau

$$\begin{aligned}
v &= 2 - 2x_1 + x_3 + z_2 \\
x_2 &= 1 - x_1 - x_3 \\
z_1 &= 2 - 3x_1 - 3x_3 + z_2
\end{aligned} \tag{2.17}$$

Similarly, the initial basis for (2.15) with basic variables u, y_2, w_2, w_3 yields the tableau

$$
\begin{aligned}
u &= 6 - 6y_1 + w_1 \\
y_2 &= 1 - y_1 \\
w_2 &= 1 - 3y_1 + w_1 \\
w_3 &= 3 - 6y_1 + w_1
\end{aligned}
\tag{2.18}
$$

which is equivalent to the equations in (2.15). A label representing a pure strategy i of player 1 is a nonbasic variable x_i or w_i, and a label representing a pure strategy j of player 2 is a a nonbasic variable y_j or z_j. Here, the $m+n$ nonbasic variables represent all labels except the missing label 2.

Label 1 is duplicate since both x_1 and w_1 are nonbasic. The move along a line segment in the algorithm results by treating x_1 as an additional variable that is no longer fixed at zero. Equation (2.17) shows how the current basic variables change when x_1 is increased. Thereby, x_2 and z_1 have to stay nonnegative, that is, $x_2 = 1 - x_1 \geq 0$ and $z_1 = 2 - 3x_1 \geq 0$ (the sign of v does not matter). These inequalities are equivalent to $x_1 \leq 1$ and $x_1 \leq {}^2\!/_3$, of which the latter is stronger. When x_1 is increased to ${}^2\!/_3$, then $z_1 = 0$. This determines a new basic solution where x_1, called the *entering* variable, is a new basic variable, and z_1, called the *leaving* variable, becomes nonbasic. This basis change is called *pivoting*. It produces a new tableau by rewriting the equation containing the leaving variable,

$$
x_1 = {}^2\!/_3 - {}^1\!/_3 z_1 - x_3 + {}^1\!/_3 z_2\,,
$$

and then substituting this expression for the entering variable x_1 in the remaining equations. The new tableau is

$$
\begin{aligned}
v &= {}^2\!/_3 - {}^2\!/_3 z_1 + 3x_3 + {}^1\!/_3 z_2 \\
x_2 &= {}^1\!/_3 + {}^1\!/_3 z_1 \phantom{{}+3x_3} - {}^1\!/_3 z_2 \\
x_1 &= {}^2\!/_3 - {}^1\!/_3 z_1 - x_3 + {}^1\!/_3 z_2
\end{aligned}
$$

and is obtained by straightforward row operations applied to the coefficients in the old tableau. This completes step III in the example.

Above, the first entering variable x_1 represents the new best response against y; letting w_1 enter the basis in (2.18) would not work since then all basic variables increase indefinitely with w_1. The next pivoting step is determined by the new duplicate label where both z_1 and y_1 are nonbasic. Since z_1 has just left the basis, y_1 is chosen as entering variable in (2.18). There, the increase of y_1 is bounded by the constraints $y_2 = 1 - y_1 \geq 0$, $w_2 = 1 - 3y_1 \geq 0$, $w_3 = 3 - 6y_1 \geq 0$, of which the second is the strongest, stating $y_1 \leq {}^1\!/_3$. Thus, w_2 leaves the basis where

$$
y_1 = {}^1\!/_3 - {}^1\!/_3 w_2 + {}^1\!/_3 w_1
$$

is substituted into the remaining equations in (2.18) to obtain the new tableau. This is the last step IV of the algorithm since w_2, which represents the missing label, has become nonbasic. The values of x and y in the current basic solution define an equilibrium since exactly one variable of each *complementary* pair x_i, w_i and y_j, z_j is nonbasic and zero, so that $x^\top w = 0$ and $y^\top z = 0$ hold, that is, (2.7) and (2.8).

We described the algorithm with alternating pivoting steps for two tableaus equivalent to (2.14) and (2.15). Since the two systems are independent, they can also be regarded as a single system of linear equations, whose bases obviously consist of the bases of the separate systems. We assume this for the general description.

Pivoting in the tableau (2.16) with an entering variable r_j for $j \notin \beta$ works with the following *minimum ratio test.* Let $c = D_\beta^{-1} b$ and $d = D_\beta^{-1} D_j$ where $c = (c_i)_{i\in\beta}$ and $d = (d_i)_{i\in\beta}$. The largest value of r_j such that $r_\beta = c - dr_j \geq \mathbf{0}$ is obviously given by

$$\min\{c_i/d_i \mid i \in \beta,\ d_i > 0\}. \tag{2.19}$$

The entering column d has at least one positive entry, so the increase of r_j is bounded, since after initialization the values of (x, y) stay in the bounded set $X \times Y$. The leaving variable r_i is given by that i in β where the minimum in (2.19) is taken. A *tie* occurs if the minimum is not unique. Then only one of at least two variables can leave the basis, but the other will have value zero after pivoting. This cannot happen in a nondegenerate game. So the leaving variable is unique, and after pivoting the new basis will be $\beta \cup \{j\} - \{i\}$.

If the leaving variable is such as to reach an equilibrium, the algorithm terminates. Otherwise, pivoting continues with the next entering variable as the *complement* of the variable that has just left the basis. Thereby, the variable x_i is called the complement of w_i and vice versa, for $1 \leq i \leq m$, and y_j is called the complement of z_j and vice versa, for $1 \leq j \leq n$. This is called the *complementary pivoting* rule. (This is, in fact, the only difference to the Simplex algorithm for solving a linear program, where the entering variable is chosen such as to improve the objective function.) In the course of the computation, one variable of each complementary pair, except the one for the missing label, is always nonbasic. Since the game is nondegenerate, the computed sequence of bases is unique and allows no repetitions (as stated in Theorem 2.6 in graph-theoretic terms), so the algorithm terminates. The details of the algorithm are straightforward.

2.6. Degenerate games

In this section, we clarify various notions of nondegeneracy used in the literature, and show that the existence of weakly but not strongly dominated strategies leads to degenerate games. Furthermore, we demonstrate how Lemke and Howson extended their algorithm to degenerate games, and discuss related issues of numerical stability.

Different authors define nondegeneracy differently, but most of these definitions are equivalent, as shown in the following theorem. There, the *support* of a mixed strategy (or any vector) x, denoted $\operatorname{supp}(x)$, is defined as the set of pure strategies that have positive probability,

$$\operatorname{supp}(x) = \{\, i \mid x_i \neq 0\}.$$

Furthermore, it is useful to assume positive expected payoffs u and v to player 1 and 2, respectively, in systems (2.14) and (2.15). This holds if A and B are non-negative and A has no zero column and B has no zero row, like in (2.10), or even simpler, if all entries of A and B are positive. We assume this without loss of generality (w.l.o.g.) since a constant can be added to all payoffs without changing the game.

Theorem 2.7. *Let (A, B) be an $m \times n$ bimatrix game with $A, B > \mathbf{0}$. Then the following are equivalent.*

(a) *The game is nondegenerate.*

(b) *Any mixed strategy x of player 1 has at most $|\operatorname{supp}(x)|$ pure best responses, and the same holds for any mixed strategy y of player 2.*

(c) *In any basic feasible solution to (2.14) and (2.15), the basic variables have positive values.*

(d) *For any x in X, the columns of the matrix $[I_m, B]$ corresponding to the labels of x are linearly independent, and for any y in Y, the rows of $\begin{bmatrix} A \\ I_n \end{bmatrix}$ corresponding to the labels of y are linearly independent.*

Proof. (a)$\Rightarrow$(b): If a mixed strategy x, say, has more than $|\operatorname{supp}(x)|$ pure best responses, then x has more than m labels, which cannot happen in a nondegenerate game according to Def. 2.5.

(b)$\Rightarrow$(c): This is similarly easy.

(c)$\Rightarrow$(d): Assume that in any basic feasible solution to (2.15), all basic variables are positive. We will show that the rows of $\begin{bmatrix} A \\ I_n \end{bmatrix}$ for the labels of any mixed strategy y are linearly independent. The same reasoning applies then to (2.14) and columns of $[I_m, B]$. Let $I = \{1, \ldots, m\}$, $N = \{1, \ldots, n\}$. Let a_i for $i \in I$ denote the rows of A, and let e_j for $j \in N$ denote the rows of I_n.

Assume that (d) is false, that is, for some y in Y, the rows a_i for $i \in K$ and e_j for $j \in L$ are linearly dependent, where $K \subseteq I$ and $L \subseteq N$ represent the sets of labels of y,

$$\begin{aligned} a_i y &= u \quad \text{for } i \in K, \qquad & a_i y &< u \quad \text{for } i \in I - K, \\ e_j y &= 0 \quad \text{for } j \in L, \qquad & e_j y &> 0 \quad \text{for } j \in N - L, \end{aligned} \tag{2.20}$$

and u is minimal subject to $\mathbf{1}_n u \geq Ay$ so $K \neq \emptyset$. Consider a linearly independent subset of these row vectors with the same span, given by the rows a_i for $i \in K'$ and

e_j for $j \in L$ where K' is a proper subset of K. For the moment, we ignore all the dependent rows a_k for $k \in K - K'$ of A and consider the smaller system, similar to (2.15),

$$\begin{aligned} \mathbf{1}_n^\top y \quad &= 1 \\ -u + a_i y \quad &= 0, \qquad i \in K' \\ -u + a_i y + w_i &= 0, \qquad i \in I - K, \end{aligned} \tag{2.21}$$

and $y, w \geq \mathbf{0}$, with variables u, y_j for $j \in N$, and w_i for $i \in I - K$. The given values for u, y, and suitable slacks w_i, solve (2.21). This system has full row rank: Otherwise, the zero vector is a nontrivial linear combination of the rows in (2.21), clearly not involving the rows for $i \in I - K$. That is, there are reals λ_0 and λ_i, $i \in K'$, not all zero, with $-\sum_{i\in K'} \lambda_i = 0$ and $\lambda_0 \mathbf{1}_n^\top + \sum_{i\in K'} \lambda_i a_i = \mathbf{0}^\top$, so that $\lambda_0 \mathbf{1}_n^\top y + \sum_{i\in K'} \lambda_i a_i y = \mathbf{0}^\top y$ and therefore $\lambda_0 = 0$, which contradicts the linear independence of the rows a_i for $i \in K'$. Since (2.21) has a feasible solution u, y, w, it has a basic feasible solution $\overline{u}, \overline{y}, \overline{w}$ with $\mathrm{supp}(\overline{y}) \subseteq \mathrm{supp}(y)$ (this is a standard, easy result on existence of basic feasible solutions).

Consider now the linear dependent rows a_k for $k \in K - K'$. Each of them is of the form

$$a_k = \sum_{i\in K'} \lambda_i a_i + \sum_{j\in L} \mu_j e_j$$

for suitable reals λ_i and μ_j, where

$$u = a_k y = \sum_{i\in K'} \lambda_i a_i y + \sum_{j\in L} \mu_j e_j y = \Big(\sum_{i\in K'} \lambda_i\Big) u$$

so $\sum_{i\in K'} \lambda_i = 1$ since $u > 0$ (because $A > \mathbf{0}$). Since $\mathrm{supp}(\overline{y}) \subseteq \mathrm{supp}(y)$, so $\overline{y}_j = e_j \overline{y} = 0$ for $j \in L$, we obtain similarly $\overline{u} = a_k \overline{y}$. The basic feasible solution $\overline{u}, \overline{y}, \overline{w}$ of (2.21) can therefore be extended to a basic feasible solution of (2.15) with zero basic variables w_k for $k \in K - K'$ (and nonbasic variables w_i for $i \in K'$), contradicting (c). This shows (c)$\Rightarrow$(d).

(d)$\Rightarrow$(a): Assume (d) holds, and let $y \in Y$ (the argument for x in X will be analogous). We use the same notation as before. The labels of y are the unique sets K and L so that (2.20) holds, $K \neq \emptyset$. Let $\overline{y} = y$, and let $d = n - |K| - |L|$. We want to find d points y in Y in addition to $\overline{y}$ fulfilling (2.20) which, together with $\overline{y}$, are affinely independent. Extend the, by assumption, linearly independent rows a_i for $i \in K$ and e_j for $j \in L$ by d rows e_j for $j \in L'$, $|L'| = d$, such that all these n rows are linearly independent. Then, for each $l \in L'$, consider the constraints in u, y

$$
\begin{aligned}
\mathbf{1}_n^\top y &= 1 && \\
-u + a_i y &= 0, && i \in K \\
-u + a_i y &< 0, && i \in I - K \\
y_j &= 0, && j \in L \\
y_j &= \overline{y}_j, && j \in L',\ j \neq l \\
y_l &= \overline{y}_l + \varepsilon && \\
y_j &> 0, && j \in N - L - L' \neq \emptyset.
\end{aligned}
$$

Similarly to (2.21), the $n+1$ equations in this system are linearly independent and determine a unique solution u and $y = y(l)$, which fulfills the inequalities if ε is sufficiently small. We can choose $\varepsilon > 0$ so that this holds for all $l \in L'$. The $d+1$ points $\overline{y}$ and $y(l)$ for $l \in L'$ have the same labels. It is easy to see that they are affinely independent, and that their affine hull contains all points y in Y that have (at least) the labels of $\overline{y}$. So the set of these points has the correct dimensionality as required in Def. 2.5. $\square$

Our Definition 2.5 of nondegeneracy in Theorem 2.7(a) is used by Krohn et al. (1991), and, in slightly weaker form, by Shapley (1974). Condition (b) is the easiest to state and should be used as definition. Van Damme (1987, p. 52) has observed the implication (d)$\Rightarrow$(b). The uniqueness of the complementary pivoting rule relies on (c). Lemke and Howson (1964) define nondegenerate games by condition (d).

A game is *not* necessarily nondegenerate if all payoffs of a player are distinct. This only guarantees (b) for pure strategies x and y. Conversely, a nondegenerate game can have identical payoffs against a pure strategy as long as these are not maximal.

Degeneracy is related to the presence of weakly dominated strategies. Consider a pure strategy k of player 1, say, with corresponding row a_k of his payoff matrix A, and a mixed strategy x of player 1. Then a_k is called *payoff equivalent* to x if $a_k = x^\top A$, *strictly dominated* by x if $a_k < x^\top A$, and *weakly dominated* by x if $a_k \leq x^\top A$ but neither of the two preceding cases hold.

Theorem 2.8. *A bimatrix game is degenerate if a player has a pure strategy which is weakly dominated by or payoff equivalent to another mixed strategy, but not strictly dominated.*

Proof. Consider a pure strategy k of player 1, say, which is weakly dominated by or payoff equivalent to a different mixed strategy. The kth row of the $m \times n$ payoff matrix A of player 1 is a_k, the entries of A are a_{ij}. Then the LP with variables $x \in \mathbb{R}^m$ and $t \in \mathbb{R}$,

$$\begin{array}{ll} \text{minimize} & -t \\ \text{subject to} & x^\top A - t\mathbf{1}_n^\top \geq a_k \\ & x_k = 0 \\ & x^\top \mathbf{1}_m = 1 \\ & x \geq \mathbf{0} \end{array}$$

has an optimal solution x, t with $t = 0$, where x is the mixed strategy that dominates or is payoff equivalent to k (if $t > 0$, then k is strictly dominated). It is easy to see that the corresponding dual optimal solution gives a mixed strategy y of player 2 against which both x and k are best responses. This implies $\sum_i x_i a_{ij} = a_{kj}$ for all j with $y_j > 0$, which violates the linear independence condition in Theorem 2.7(d) for the rows of $\begin{bmatrix} A \\ I_n \end{bmatrix}$ corresponding to the labels of y (w.l.o.g., $A > \mathbf{0}$). So the game is degenerate. $\square$

Iterated elimination of weakly dominated and payoff equivalent pure strategies does not necessarily yield a nondegenerate game, however (like the game in (2.30) below). Furthermore, this elimination process is not always desired since it may produce different games depending on the order of elimination, as in the game

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}, \tag{2.22}$$

where eliminating first the bottom row and then the right column yields a different game than eliminating first the left column and then the top row, for example. (Knuth, Papadimitriou, and Tsitsiklis, 1988, study computational aspects of strategy elimination where they overlook this fact.)

It is often assumed that a game is nondegenerate on the grounds that this is true for a *generic* game. A generic game is a game where the payoffs are not exactly known. We use the following definition.

Definition 2.9. *A game is called generic if each payoff is drawn randomly and independently from a continuous distribution.*

A game is generic, for example, if each payoff a_{ij} for player 1 (similarly b_{ij} for player 2) is drawn independently from the uniform distribution on the interval $[\overline{a}_{ij} - \varepsilon, \overline{a}_{ij} + \varepsilon]$ for some $\varepsilon > 0$, where $\overline{a}_{ij}$ represents an approximate value for the respective payoff.

By Theorem 2.7(d), a sufficient condition that the game is nondegenerate is that *any* n rows of the matrix $\begin{bmatrix} A \\ I_n \end{bmatrix}$ (not just those corresponding to the labels of y in Y) are linearly independent, and that the same holds for the columns of the matrix $[I_m, B]$. The rows of A are then said to be in *general position.* For a generic game, this is true with probability one, since a linear dependence imposes an equation on at least one payoff, which is fulfilled with probability zero. So a generic game is nondegenerate (with probability one). The common understanding of a "generic game" is that its payoffs are realizations of random variables as stated in Def. 2.9, where some undesirable, zero probability event (here degeneracy) does not occur. This is reasonable and convenient since a precise statement of the undesired conditions can be laborious, as we have seen here.

While degeneracy can be excluded due to accidental causes, it may well occur for *systematic* reasons, if the entries of the payoff matrix are not independent. In particular, this holds if the bimatrix game is the normal form of a nontrivial extensive form game, where there are more strategy combinations than payoffs. Fortunately, the Lemke–Howson algorithm can easily be extended to degenerate games.

A degenerate game has basic feasible solutions to (2.14), (2.15) where some basic variables have value zero, by Theorem 2.7(c). Such a *degenerate basis* is computed by the Lemke–Howson algorithm after a pivoting step where the leaving variable, determined by the minimum ratio test (2.19), is not unique. In graph-theoretic terms, this basis represents a vertex where more than two almost completely labeled edges meet, so Theorem 2.6 is no longer valid. Instead of following a unique path, the computation must choose a branch and may possibly cycle.

The solution to this problem is the so-called *lexicographic* rule for choosing the leaving variable in the case of ties, which is well known in linear programming (see, for example, Chvátal, 1983, p. 36), and has been suggested by Lemke and Howson (1964) in terms of perturbations. For a general lexicographic treatment of LCPs see Eaves (1971). Consider a system $Dr = b$ of k linearly independent equations with the vector r of variables subject to $r \geq \mathbf{0}$. *Perturb* the right hand side b by replacing it by $b(\varepsilon) = b + (\varepsilon, \ldots, \varepsilon^k)^\top$ for some $\varepsilon > 0$. For a basis β of this system, the vector $r_\beta = (r_i)_{i\in\beta}$ of basic variables is then given by $r_\beta = D_\beta^{-1} b(\varepsilon) = D_\beta^{-1} b + D_\beta^{-1}(\varepsilon, \ldots, \varepsilon^k)^\top$. Let $(c_{i0}, c_{i1}, \ldots, c_{ik})$ denote row i of the matrix $D_\beta^{-1}[b, I_k]$, so that $r_i = c_{i0} + c_{i1}\varepsilon + \cdots + c_{ik}\varepsilon^k$, for $i \in \beta$. If the first nonzero entry of $(c_{i0}, c_{i1}, \ldots, c_{ik})$ is positive, then $r_i > 0$ if ε is sufficiently small. If this holds for all rows i of $D_\beta^{-1}[b, I_k]$, this matrix is called *lexico-positive*, and β is called *lexico-feasible* since then β is feasible for the original system $Dr = b$. Moreover, a lexico-feasible basis β is feasible and *nondegenerate* for the perturbed system $Dr = b(\varepsilon)$, even if it is degenerate for $Dr = b$.

Thus, perturbation of the system guarantees that every computed basis is nondegenerate. The point of lexico-feasible bases β is that they can be recognized from $Dr = b$, via D_β^{-1}, without perturbing the system at all. By suitable rules, the perturbation for $\varepsilon > 0$ is *simulated*, and the computed basic solutions are not changed, as if ε is vanishing. With the initialization of the algorithm, the first feasible basis can easily be chosen such that it is lexico-feasible (which holds directly if it is nondegenerate). Consider the pivoting step with entering column $d = D_\beta^{-1} D_j$ in (2.16) and entering variable r_j. The leaving variable is determined by the maximum choice of r_j such that, for the perturbed system,

$$r_i = c_{i0} + c_{i1}\varepsilon + \cdots + c_{ik}\varepsilon^k - d_i r_j \geq 0$$

for all $i \in \beta$. Assuming ε is sufficiently small, the sharpest bound is obtained for that i with the *lexicographically smallest* of the row vectors $1/d_i \cdot (c_{i0}, c_{i1}, \ldots, c_{ik})$ for $d_i > 0$ (a vector is called lexicographically smaller than another if it is smaller in the first component where the vectors differ). That is, if there is a tie among the smallest ratios c_{i0}/d_i for $d_i > 0$, then among these the smallest ratio c_{i1}/d_i, and if there is a tie then among them the smallest ratio c_{i2}/d_i, and so on, determines the leaving variable. No two of these row vectors are equal since $D_\beta^{-1}[b, I_k]$ has full rank k. Therefore, this rule determines the leaving variable r_i uniquely. It is called the *lexico-minimum ratio test* and extends (2.19). By construction, it preserves the invariant that all computed bases are lexico-positive. Since the computed sequence of bases is unique, no basis can be repeated and the algorithm terminates as in the nondegenerate case.

With a variant of the lexicographic rule, the Lemke–Howson algorithm can be used to compute equilibria with additional stability properties (Wilson, 1992). We will explain this in Section 3.1.

The practical relevance of a systematic treatment of degeneracy is less clear, however. For example, degenerate bases are frequent when sparse linear programs are solved with the Simplex algorithm, but cycling is extremely rare. *Numerical stability* is a much more important issue when computing with floating point arithmetic, since roundoff errors can be amplified in pivoting steps. Compared to the Simplex algorithm, this problem is particularly severe for the complementary pivoting rule (Tomlin, 1978; Cottle et al., 1992, p. 383).

Moreover, the lexicographic rule is questionable in this context. In floating point arithmetic, computed numbers are often treated as equal if their difference does not exceed a certain small tolerance. This may randomly lead to degeneracies where the lexicographic rule loses its foundation since it simulates perturbing the data with numbers ε^i that are much smaller than the tolerance. The best way out of this seems to work with *rational arithmetic* where numbers are always represented exactly as fractions of integers. This method is slow, but it seems suitable for integer payoff matrices, and avoids the problem of numerical instability.

2.7. Finding all equilibria

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. The problem of finding all equilibria can be phrased as a vertex enumeration problem for polytopes.

We first recall some notions from polytope theory (where we recommend Ziegler, 1995). A *polyhedron* H is a subset of $\mathbb{R}^d$ defined by a finite number of linear inequalities. If the dimension of H is d, then H is called *full-dimensional.* A *polytope* is a bounded polyhedron. A *face* of a polyhedron H is a subset of H of the form $\{ x \in H \mid c^\top x = c_0 \}$ where $c^\top x \leq c_0$ is a valid inequality for H (holds for all x in H), for some $c \in \mathbb{R}^d$, $c_0 \in \mathbb{R}$. A face of dimension zero (or its unique element) is called *vertex* of H. This is the same as an *extreme point* of H, that is, a point in H not representable as a convex combination of other points in H. A face of dimension one is called an *edge* of H. A face of dimension $d-1$ is called a *facet* of H if H has dimension d. If $H = \{ x \in \mathbb{R}^d \mid Ax \leq b \}$ for some matrix A and vector b, and a_i is a row of A and b_i the corresponding component of b, and $x \in H$, then the inequality $a_i x \leq b_i$ is called *binding* for x if $a_i x = b_i$. It can be shown that any nonempty face of H can be characterized by a set of binding inequalities, that is, by turning some of the inequalities defining H into equalities. Each 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).

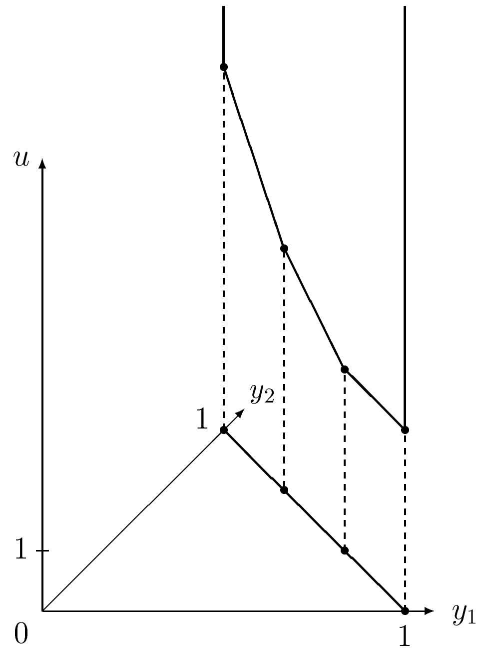

Figure 2.4. The polyhedron H_2 for the game in (2.10), and its projection to the set $\{ (y,0) \mid (y,u) \in H_2 \}$. The vertical scale is displayed shorter.

Let (A,B) be an $m \times n$ bimatrix game, and consider the polyhedra

$$H_1 = \{(x,v) \mid \mathbf{1}_m^\top x = 1,\ B^\top x \leq \mathbf{1}_n v,\ x \geq \mathbf{0}\}$$

and

$$H_2 = \{(y,u) \mid \mathbf{1}_n^\top y = 1,\ Ay \leq \mathbf{1}_m u,\ y \geq \mathbf{0}\}\,.$$

The elements of $H_1 \times H_2$ describe the solutions to (2.9), or equivalently to (2.14) and (2.15) when using the vectors of slack variables z and w. Figure 2.4 shows a drawing of H_2 for our example (2.10). For (y, u) in H_2, the vector y (drawn in the horizontal plane) is constrained by $1^\top y = 1$ and $y \geq \mathbf{0}$ which defines the set Y of mixed strategies of player 2 as above. The scalar u (drawn vertically) is at least the maximum of the functions $a_i y$ for the rows a_i of A. The maximum itself (defining the "upper envelope" of these functions) 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 by the map $(y, u) \mapsto (y, 0)$) reveals the subdivision of Y into best response regions similar to Figure 2.2. We observe here that the (maximally labeled) points marked by dots appear as projections of the vertices of H_2.

The sets H_1 and H_2 have dimension m and n, respectively, which is one dimension higher than the dimensions $m-1$ and $n-1$ of the respective strategy spaces X and Y. If m and n are at most four, then the subdivisions of X and Y can be visualized. The polyhedra H_1 and H_2, however, are in general simpler to study than the subdivided strategy spaces. In fact, the graphs G_1 and G_2 in Theorem 2.6, as far as they are subsets of X and Y, consist of the vertices and edges of H_1 and H_2 (disregarding the components v and u, which are only used in the algebraic definition of the Lemke–Howson algorithm).

We want to simplify H_1 and H_2 further. If $(x, u) \in H_1$ and $(y, v) \in H_2$, then v and u can be arbitrarily large. Thus, the polyhedra H_1 and H_2 are unbounded. Furthermore, they are not full-dimensional. As before, we assume w.l.o.g. $A > \mathbf{0}$ and $B > \mathbf{0}$ to guarantee $u > 0$ and $v > 0$ (adding a constant to all payoffs just increases u and v by that constant, so H_1 and H_2 keep their shape). Let

$$P_1 = \{\, x' \in \mathbb{R}^m \mid B^\top x' \leq \mathbf{1}_n,\ x' \geq \mathbf{0}\} \tag{2.23}$$

and

$$P_2 = \{\, y' \in \mathbb{R}^n \mid Ay' \leq \mathbf{1}_m,\ y' \geq \mathbf{0}\}\,. \tag{2.24}$$

These polyhedra are polytopes and full-dimensional since $A > \mathbf{0}$ and $B > \mathbf{0}$.

The set H_1 is in one-to-one correspondence with $P_1 - \{\mathbf{0}\}$ with the map $(x, v) \mapsto x \cdot (1/v)$, and 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

$$x = x' \cdot v, \quad v = 1/\mathbf{1}_m^\top x', \qquad y = y' \cdot u, \quad u = 1/\mathbf{1}_n^\top y'. \tag{2.25}$$

These bijections are not linear. However, each defines a *projective transformation* (see Ziegler, 1995, Sect. 2.6) which maps faces to faces. This is clear since a binding inequality in H_1 corresponds to a binding inequality in P_1 and vice versa. Figure 2.5 shows a geometric interpretation of the bijection $(y, u) \mapsto y \cdot (1/u)$. 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

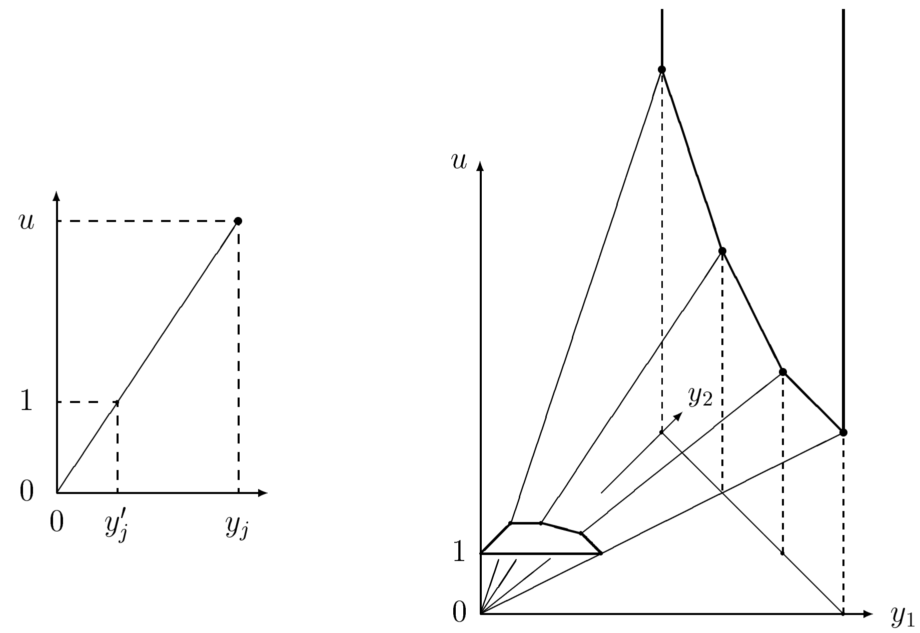

Figure 2.5. The map $H_2 \to P_2$, $(y, u) \mapsto y' = y \cdot (1/u)$ as a projective transformation from $\mathbb{R}^{n+1}$ to the hyperplane $\{(y', 1) \mid y' \in \mathbb{R}^n\}$ with projection point $(\mathbf{0}, 0)$. The left hand side shows this for a single component y_j of y, the right hand side shows how P_2 arises in this way from H_2 in the example (2.10).

line connecting this pair to $(0, 0)$ contains the point $(y'_j, 1)$ with $y'_j = y_j/u$. Thus, P_2 can be seen as the intersection of the lines connecting any (y, u) in H_2 with $(\mathbf{0}, 0)$ in $\mathbb{R}^{n+1}$ with the set of points $(y', 1)$, as shown on the right hand side in Figure 2.5 for our example. The point $\mathbf{0}$ in P_2 does not arise as such a projection, but can be thought of as corresponding to a ficticious vertex of H_2 "at infinity".

The inequalities defining P_1 and P_2 are simpler than those for H_1 and H_2. The Lemke–Howson algorithm can be described for these inequalities converted to equality form, starting from the point $(\mathbf{0}, \mathbf{0})$ in $P_1 \times P_2$ as an artificial equilibrium similar to the one used in Section 2.6. This is often done to simplify the algebraic computation, in particular the initialization step, but requires the retranslation of the computed vectors (x', y') to mixed strategy pairs (x, y) and payoffs v, u that we are using now.

We want to show that finding all equilibria of (A, B) can be reduced to an inspection of all vertices of P_1 and P_2. The points x' in $P_1 - \{\mathbf{0}\}$ and y' in $P_2 - \{\mathbf{0}\}$ are always translated to the corresponding mixed strategies x and y, respectively, according to (2.25). As before, the rows of A are a_i for $1 \le i \le m$, and the columns of B are B_j for $1 \le j \le n$. To identify equilibria, we use again labels, that is, elements of $\{1, \ldots, m + n\}$ denoting own pure strategies that have probability zero or pure best responses of the opponent. The labels of x' in P_1 and of y' in P_2 are the same as those of x and y, respectively, with (2.25). They are given by

$$\begin{aligned} L(x') &= \{\, i \mid x'_i = 0 \,\} \cup \{ m+j \mid B_j^\top x' = 1 \,\}, \\ L(y') &= \{\, i \mid a_i y' = 1 \,\} \cup \{ m+j \mid y'_j = 0 \,\}. \end{aligned} \tag{2.26}$$

Labels represent binding inequalities. The points with a maximal number of linearly independent binding inequalities are the vertices.

Theorem 2.10. *Consider P_2 as in (2.24), for $A > \mathbf{0}$. Then the following are equivalent:*

(a) y' is a vertex of P_2,

(b) $y' \in P_2$, and y' fulfills n linearly independent equalities of the form $a_i y' = 1$ or $y'_j = 0$,

(c) y', w is a basic feasible solution of the system $Ay' + I_m w = \mathbf{1}_m$, $y, w \geq \mathbf{0}$, with $w = \mathbf{1}_m - Ay'$.

Proof. Exemplarily, we show (b)⇒(a). If y' has n linearly independent binding inequalities, then y' is uniquely determined, and an extreme point of P_2 since an inequality is binding for a convex combination of points in P_2 iff it is binding for all these points. The other implications are similarly easy. □

This theorem, which holds analogously for P_1, does not require that the game is nondegenerate. If the game is nondegenerate, then by Theorem 2.7(d), the binding inequalities for any point in P_1 or P_2 are always independent. Thus, by Theorem 2.10, every vertex of P_1 or P_2 has exactly m respectively n binding inequalities, and only vertices are part of equilibria. Then, we can find all equilibria as follows.

Theorem 2.11. *Let (A, B) be a nondegenerate $m \times n$ bimatrix game, $A, B > \mathbf{0}$. The set of all equilibria of (A, B) can be computed as follows. Enumerate the vertices x' of the polytope P_1 in (2.23) and the vertices y' of the polytope P_2 in (2.24). Then (x, y) is an equilibrium iff, with (2.25) and (2.26), $L(x') \cup L(y') = \{1, \ldots, m+n\}$ and $(x', y') \neq (\mathbf{0}, \mathbf{0})$.*

This method was first suggested by Vorob'ev (1958) and later simplified by Kuhn (1961). Mangasarian (1964) inspects the vertices of $H_1 \times H_2$ to check for maxima of the bilinear function $x^\top(A+B)y - u - v$. At a maximum, this function is zero, so this is equivalent to the complementarity conditions (2.7) and (2.8), that is, completely labeled vertices. Enumerating the vertices of a polytope given by linear inequalities is a standard problem (see Ziegler, 1995). A recent elegant method, which has apparently not yet been applied to bimatrix games, is due to Avis and Fukuda (1992). There, the inequalities of the polytope, say P_2, are converted to a system of equations as in Theorem 2.10(c), whose basic feasible solutions are enumerated by a depth-first search based on the Simplex algorithm (with a unique pivoting rule) "run backwards". Audet et al. (1996) present a different approach that restricts the search of vertices guided by the equilibrium condition. In contrast, Dickhaut and Kaplan (1991) ignore the polytope structure and enumerate all possible supports of equilibria.

This raises the question if the polytopes provide useful information, rather than complicating the algorithm. It is known that the number of vertices of a polytope specified by inequalities can grow exponentially with the dimension. For example, the d-dimensional unit cube is defined by $2d$ inequalities and has 2^d vertices. This is not the largest possible number, however. The upper bound is obtained for the so-called dual neighborly polytopes, which have the number of vertices stated in the following result.

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

$$\Phi(d,k) = \binom{k - \lfloor \frac{d-1}{2} \rfloor - 1}{\lfloor \frac{d}{2} \rfloor} + \binom{k - \lfloor \frac{d}{2} \rfloor - 1}{\lfloor \frac{d-1}{2} \rfloor}. \tag{2.27}$$

In (2.27), $\binom{p}{q}$ is the binomial coefficient $p!/(q!(p-q)!)$ and $\lfloor r \rfloor$ denotes the largest integer not exceeding r, for natural numbers p, q and real r. For a self-contained proof of this theorem see Mulmuley (1994). The representations of $\Phi(d,k)$ in terms of binomial coefficients vary in the literature. They are easily converted into each other using the arrangement of binomial coefficients in the Pascal triangle according to their property $\binom{p}{q} = \binom{p-1}{q-1} + \binom{p-1}{q}$. In particular, one can rewrite (2.27) depending on whether d is even, $d = 2p$, or odd, $d = 2p+1$:

$$\Phi(2p,k) = \frac{k}{p}\binom{k-p-1}{p-1}, \qquad \Phi(2p+1,k) = 2\binom{k-p-1}{p}. \tag{2.28}$$

The upper bound theorem shows that P_1 has at most $\Phi(m, n+m)$ and P_2 at most $\Phi(n, m+n)$ vertices, including $\mathbf{0}$ which is not part of an equilibrium. The numbers $2^m - 1$ and $2^n - 1$ of possible supports for mixed strategies grow much faster, so it is advisable to use the method of Theorem 2.11.

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. That is, Theorem 2.12 implies the following.

Corollary 2.13. *(Keiding, 1997.) A nondegenerate 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 $\Phi(m, n+m) < \Phi(n, m+n)$. The case $m = n = d$, that is, a square bimatrix game, has received special attention. In that case, one can use the single expression

$$\Phi(d, 2d) = 2\binom{\lfloor \frac{3d-1}{2} \rfloor}{\lfloor \frac{d-1}{2} \rfloor} \tag{2.29}$$

which follows form (2.28) with $k/p = 2 \cdot 2p/p$ if $d = 2p$. For $d = 1, \ldots, 6$, $\Phi(d, 2d)$ equals $2\binom{1}{0}$, $2\binom{2}{1}$, $2\binom{4}{1}$, $2\binom{5}{2}$, $2\binom{7}{2}$, $2\binom{8}{3}$, that is, 2, 4, 8, 20, 42, 112, respectively. However, the $d \times d$ bimatrix games with the largest of number of equilibria known are those where both players have the identity matrix as payoff matrix. In that case, the game has $2^d - 1$ equilibria, namely for any $\emptyset \neq C \subseteq \{1, \ldots, d\}$ the mixed strategy pairs (x, y) where both x and y play every pure strategy in C with probability $1/|C|$. Then both P_1 and P_2 are equal to the unit cube. Possibly, this is a tighter bound for the number of equilibria than $\Phi(d, 2d) - 1$.

Conjecture 2.14. *(Quint and Shubik, 1994.) A nondegenerate $d \times d$ bimatrix game has at most $2^d - 1$ equilibria.*

This conjecture is a consequence of Corollary 2.13 for $d \leq 3$ but not for $d > 3$. For $d = 4$, it was shown by Keiding (1997) and McLennan and Park (1996). Nevertheless, potential counterexamples may arise in higher dimensions since $\Phi(d, 2d)$ grows faster with d than 2^d. [This is indeed the case. Using the polytope approach, the Quint–Shubik conjecture has been disproven for $d = 6$ and all $d \geq 8$ by von Stengel (1997).] As the small values for d show, $\Phi(d, 2d)$ increases, by (2.29), from even to odd d with an asymptotic factor of 9/4, and from odd to even d with an asymptotic factor of 3, so that $\Phi(d, 2d)$ is asymptotically about $(27/4)^{d/2}$ or approximately 2.6^d.

We return to the problem of finding all equilibria of an arbitrary, possibly degenerate bimatrix game. Interpreted for the polytopes P_1 and P_2, degeneracy has two possible reasons. The first is a redundancy of the *description* of the polytope. For a strictly dominated strategy, say strategy k of player 1, the inequality $a_k y \leq 1$ for P_2 is never binding, so this inequality is redundant and can be omitted without changing P_2. If the strategy is weakly dominated by or payoff equivalent to another mixed strategy but not strictly dominated, then the equality is sometimes binding and the game is degenerate according to Theorem 2.8. Nevertheless, such an inequality is still redundant, that is, it does not define a facet, and it can be omitted without changing the polytope. The second reason for degeneracy can be recognized from the polytope itself. Assume that each inequality defines a facet. Then in a degenerate game, P_1 or P_2 has a vertex that belongs to more than d facets, where d is the dimension (m or n) of the polytope. A polytope where each vertex belongs to exactly d facets is called *simple*. 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 $(0, 0, 1/4)^\top$ belongs to four facets. In summary, a game is degenerate because of weakly but not strictly dominated strategies as in Theorem 2.8, or because P_1 or P_2 is not a simple polytope.

Theorem 2.11 can be modified slightly to obtain a method for enumerating the equilibria of any game, as shown by Mangasarian (1964); see also Winkels (1979).

Theorem 2.15. *Let (A, B) be an $m \times n$ bimatrix game, $A, B > \mathbf{0}$. The set of all equilibria of (A, B) can be computed as follows. Enumerate the vertices x' of the polytope P_1 in (2.23) and the vertices y' of the polytope P_2 in (2.24), and call x' a mate of y' iff $L(x') \cup L(y') = \{1, \ldots, m + n\}$ according to (2.26). Let a clique be any set of vertex pairs (x', y') where x' is a mate of y'. Then (x, y) is an equilibrium iff (2.25) holds and (x', y') belongs to the convex hull of a maximal clique, and $(x', y') \neq (\mathbf{0}, \mathbf{0})$.*

Proof. First, any point x' in P_1 is a mate of $\mathbf{0}$ in P_2 iff $\{1, \ldots, m\} \subseteq L(x')$, that is, $x' = \mathbf{0}$. Similarly, $\mathbf{0}$ in P_1 is a mate of y' in P_2 iff $y' = \mathbf{0}$. Thus, $\mathbf{0}$ is not part of any pair of mates (x', y') except $(\mathbf{0}, \mathbf{0})$. These pairs $(x', y') \neq (\mathbf{0}, \mathbf{0})$ can be translated by (2.25) to elements of $H_1 \times H_2$ and represent equilibria (x, y).

Let $(\overline{x}, \overline{y})$ be any equilibrium of the game, with payoffs $\overline{v}, \overline{u}$. Fixing $y = \overline{y}$ and $u = \overline{u}$, the constraints defining H_1 and the complementarity conditions (2.7) and (2.8) are linear in x, v. They define a polyhedron that may not be a singleton if the game is degenerate. In particular, (2.7) is equivalent to $x_i = 0$ for $i \in \mathrm{supp}(\mathbf{1}_m\overline{u} - A\overline{y})$ and $x^\top B_j = v$ for $j \in \mathrm{supp}(\overline{y}) \neq \emptyset$. The set of (x, v) fulfilling these constraints is bounded, so it is a polytope and equal to the convex hull of its vertices, which are vertices of H_1 since the set is a face of H_1. So $(\overline{x}, \overline{v})$ is a convex combination of these vertices. We consider only vertices (x^s, v^s) that are used for the convex combination (for example, $(\overline{x}, \overline{v})$ may itself be a vertex of H_1). That is, $\overline{x} = \sum_{s \in S} x^s \lambda_S$ and $\overline{v} = \sum_{s \in S} v^s \lambda_s$ with $\lambda_s > 0$ for $s \in S$ and $\sum_{s \in S} \lambda_s = 1$. Similarly, $(\overline{y}, \overline{u})$ is a convex combination of extreme points (y^t, u^t) of H_2 such that $(\overline{x}, y^t)$ is an equilibrium. Again, we consider only the representation $\overline{y} = \sum_{t \in T} y^t \mu_t$ and $\overline{u} = \sum_{t \in S} u^t \mu_t$ where $\mu_t > 0$ for $t \in T$ and $\sum_{t \in T} \mu_t = 1$.

Let $s \in S$ and $t \in T$. We claim (x^s, y^t) is an equilibrium. Namely, if $1 \leq i \leq m$, then $x_i^s > 0$ implies $\overline{x}_i > 0$ since $\lambda_s > 0$, thus $a_i\overline{y} = \overline{u}$ and therefore $a_i y^t = u^t$ since $\mu_t > 0$ (note $a_i y^t < u^t \implies a_i\overline{y} < \overline{u}$). Similarly, $1 \leq j \leq n$ and $y_j^t > 0$ imply that j is a best response to x^s. So

$$(\overline{x}, \overline{y}) = \sum_{s \in S,\ t \in T} \lambda_s \mu_t (x^s, y^t)$$

with the extreme equilibria (x^s, y^t), which together with their payoffs v^s, u^t correspond to extreme points (x', y') of $P_1 \times P_2$ where x' is a mate of y'. These form a clique of mates, which is a subset of a maximal clique.

Conversely, it is immediate that convex combinations of pairs (x', y') of mutual mates correspond to equilibria. $\square$

Theorem 2.15 shows that when looking for equilibria, it suffices to look for extreme equilibria corresponding to pairs of vertices of P_1 and P_2. Furthermore, it shows which convex combinations of these extreme equilibria are further equilibria. Maximal convex subsets of $X \times Y$ of equilibria are called *convex equilibrium components*, which are not necessarily disjoint. They have been characterized by Jansen (1981). Theorem 2.15 shows that the sets of their vertices can be computed as maximal cliques of a bipartite graph. As an example, the game in (2.22) has the pairs of extreme equilibria $\big((1,0)^\top, (1,0)^\top\big)$, $\big((1,0)^\top, (0,1)^\top\big)$, and $\big((0,1)^\top, (0,1)^\top\big)$. The first two and the last two of these pairs each form a maximal clique. The convex equilibrium components of this game are therefore $\{(1,0)^\top\} \times Y$ and $X \times \{(0,1)^\top\}$.

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 (see van Damme, 1987, for a survey and comparison). From the computational viewpoint, we will explain two approaches that extend the Lemke–Howson method to finding equilibria with additional refinement properties. This is the algorithm by Wilson (1992) for finding *simply stable* equilibria, and the complementary pivoting method due to van den Elzen and Talman (1991) for finding a *perfect* equilibrium.

3.1. Simply stable equilibria

Kohlberg and Mertens (1986) proposed a concept of strategic *stability* of equilibria. Basically, a set of equilibria is called stable if every game nearby has equilibria nearby (Wilson, 1992). In a nondegenerate game, all equilibria are isolated and determined by the set of pure strategies that have positive probability, where the others have strictly smaller payoff. This qualitative property does not change if the data of the game are slightly changed. So in a nondegenerate game, all (sets of) equilibria are stable.

In a degenerate game, however, stability is no longer guaranteed. The bimatrix game (A, B) in (2.30), for example, has mixed strategy sets X and Y with labels denoting best responses and unplayed pure strategies shown in Figure 3.1. This game is degenerate since $x^1 = x^2 = (0,0,1)^\top$ has four labels $1,2,4,5$, so that any point y in Y labeled 3 combined with x^1 is an equilibrium. Therefore, all convex combinations of (x^1, y^1) and (x^2, y^2), where $y^1 = (0,1)^\top$ and $y^2 = ({}^1\!/\!_3, {}^2\!/\!_3)^\top$, are equilibria. If B is changed so that

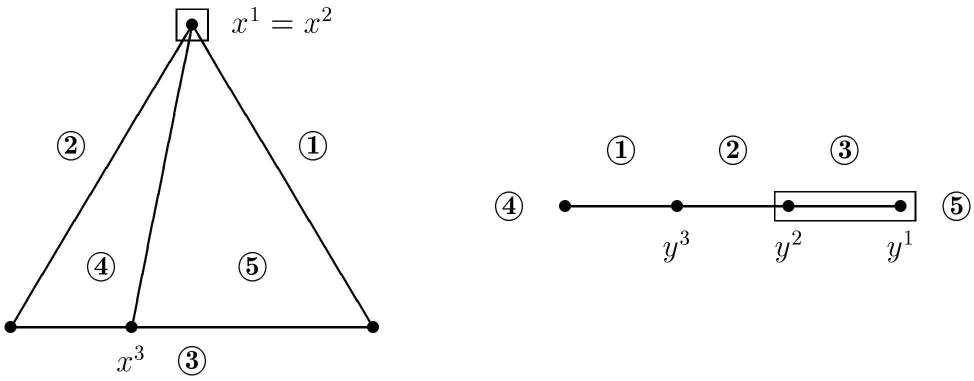

Figure 3.1. The sets X and Y of mixed strategies for the game (A, B) in (2.30), labeled with best response regions and unused pure strategies $1, 2, 3$ for player 1 and $4, 5$ for player 2. They are embedded in the next higher dimension as in Figure 2.2. The game is degenerate and has an infinite set of equilibria indicated by the rectangular boxes, with extreme equilibria (x^1, y^1) and (x^2, y^2).

$$B = \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 4 & 4+\varepsilon \end{bmatrix}, \tag{3.1}$$

then for $\varepsilon \neq 0$ this perturbed game has a unique best response to $(0, 0, 1)^\top$, namely the second pure strategy of player 2 (label 5) if $\varepsilon > 0$ and the first pure strategy (label 4) if $\varepsilon < 0$. Figure 3.2 shows that for $\varepsilon > 0$, the equilibrium component disappears (so it is not stable), whereas for $\varepsilon < 0$ it turns into two separate equilibria (x^1, y^1) and (x^2, y^2) with $x^1 \neq x^2$.

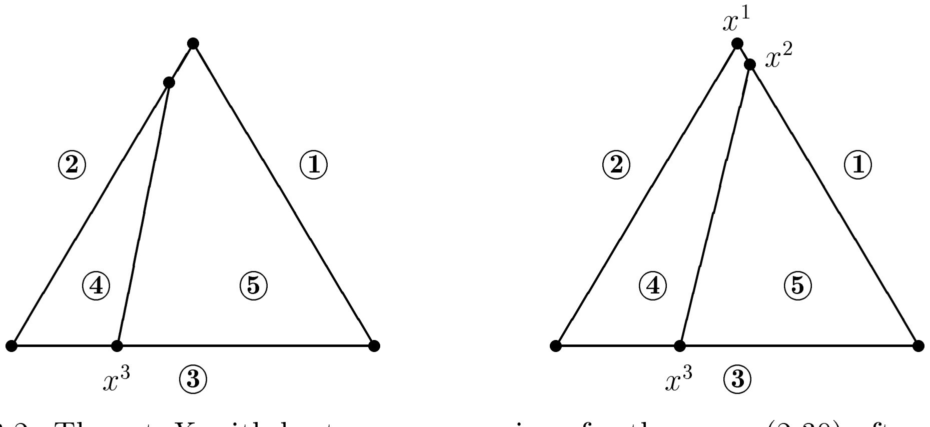

Figure 3.2. The set X with best response regions for the game (2.30) after perturbation as in (3.1), for $\varepsilon > 0$ and $\varepsilon < 0$. For $\varepsilon > 0$, the labels show that the equilibrium component with extreme equilibria (x^1, y^1), (x^2, y^2) in Figure 3.1 disappears, whereas for $\varepsilon < 0$ it dissolves into these two separate equilibria.

Here, the fact that an even number of equilibria appears indicates that another perturbation may produce zero equilibria, since the total number of equilibria in a nondegenerate bimatrix game (if such a game is obtained after perturbation) is odd according to Theorem 2.6. As a very informal proof that at least one stable component exists, observe that for that reason not all components can dissolve into an even number of equilibria, so at least one component has an odd (and therefore positive) number of equilibria remaining. Of course, this is far from a precise statement, since, among other things, convex equilibrium components need not be disjoint and perturbation does not necessarily lead to a nondegenerate game. The formal proofs in Kohlberg and Mertens (1986), or in the reformulation of stability in Mertens (1989, 1991), involve homotopies of deformations of games.

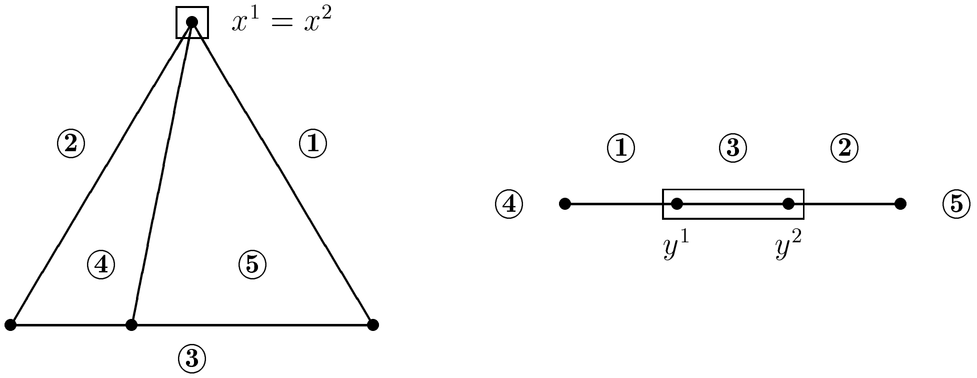

Figure 3.3. The game (3.2) has a single component, which for $\varepsilon > 0$ and $\varepsilon < 0$ in (3.1) becomes one of its extreme equilibria (x^1, y^1) or (x^2, y^2), respectively, so no equilibrium survives the perturbation individually.

To get existence, stability has to be defined for sets of equilibria. The game (A, B) with

$$A = \begin{bmatrix} 0 & 6 \\ 3 & 3 \\ 2 & 5 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 4 & 4 \end{bmatrix} \tag{3.2}$$

has the single set of equilibria with extreme points (x^1, y^1) and (x^2, y^2) shown in Figure 3.3. After perturbing B as in (3.1), only one of these equilibria (with x slightly changed) remains. Therefore, stability holds only for the entire set but not for single equilibria.

Degenerate normal form games are important because they arise from games in extensive form. Figure 3.4 shows an extensive game that has the game in (2.30) as normal form. (Extensive form games will be considered in more detail below.) The degeneracy that arises is typical.

Wilson (1992) described an algorithm that computes a set of equilibria which is stable in a weaker sense than described by Kohlberg and Mertens (1986) or the

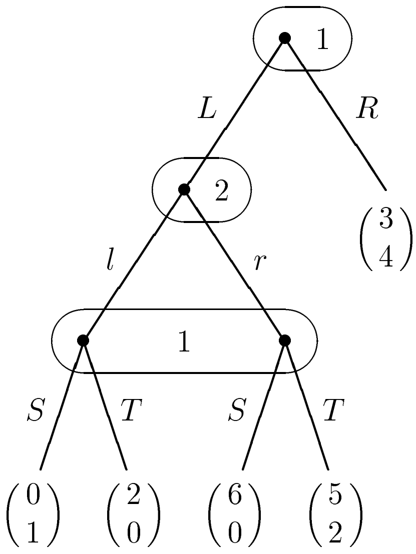

Figure 3.4. An extensive game that has the degenerate normal form game (2.30), even though the payoffs in the extensive game can be generic. The pure strategies of player 1 with labels $1, 2, 3$ in Figure 3.1 correspond to the move combinations $\langle L, S\rangle$, $\langle L, T\rangle$, and $\langle R, *\rangle$ ($*$ being arbitrary). The two pure strategies of player 2 are his moves l and r.

even stronger definition in Mertens (1989, 1991). That is, the game is not perturbed arbitrarily but only in certain systematic ways that are easily captured computationally. Because there is a smaller set of "games nearby" than in the other notions of stability, the resulting concept is weaker and is called *simple stability.* There may be simply stable sets of equilibria which are not stable, although no counterexample 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).

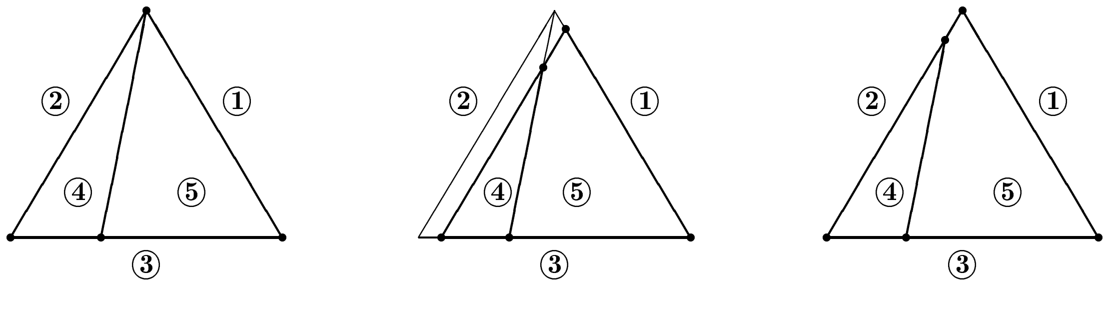

Figure 3.5. Example of a mixed strategy set for player 1 with the unperturbed game (left), a primal perturbation with a minimum probability for the strategy labeled 2 of player 1 (middle), and a dual perturbation with a bonus for the strategy labeled 5 of player 2 (right).

The perturbations considered for simple stability apply not to 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. Figure 3.5 shows the effect of such perturbations on a strategy set and its best response regions.

For computation, Wilson (1992) uses the simplified version of the LCP with inequalities as in (2.23) and (2.24). Assume w.l.o.g. that the $m \times n$ payoff matrices A and B are positive. For $x' \in \mathbb{R}^m$ and $y' \in \mathbb{R}^n$ not equal to $\mathbf{0}$, the conditions

$$\begin{array}{rcl} & Ay' & \leq \mathbf{1}_m \\ B^\top x' & & \leq \mathbf{1}_n \\ & x', \quad y' & \geq \mathbf{0} \end{array} \tag{3.3}$$

are equivalent to (2.9) using the correspondence (2.25), and the equations

$$x'^\top(\mathbf{1}_m - Ay') = 0, \qquad y'^\top(\mathbf{1}_n - B^\top x') = 0 \tag{3.4}$$

are equivalent to (2.7), (2.8). With slack vectors $w \in \mathbb{R}^m$ and $z \in \mathbb{R}^n$, the LCP (3.3), (3.4) can also be written as

$$\begin{array}{rrrl} & Ay' + I_m w & & = \mathbf{1}_m \\ B^\top x' & & + I_n z & = \mathbf{1}_n \end{array} \tag{3.5}$$

with $x', y', w, z \geq \mathbf{0}$, and

$$x'^\top w = 0, \qquad y'^\top z = 0\,. \tag{3.6}$$

Let x', y', w, z be perturbed by corresponding vectors ξ, η, ψ, ζ that have small positive components, $\xi, \psi \in \mathbb{R}^m$ and $\eta, \zeta \in \mathbb{R}^n$. That is, replace (3.5) by

$$\begin{array}{rrrl} & A(y'+\eta) + I_m(w+\psi) & & = \mathbf{1}_m \\ B^\top(x'+\xi) & & + I_n(z+\zeta) & = \mathbf{1}_n. \end{array} \tag{3.7}$$

If (3.7) and the complementarity condition (3.6) hold, then a variable x_i or y_j that is zero is replaced by ξ_i or η_j, respectively. After the transformation (2.25), that is, multiplication with u or v, these terms denote a small positive probability for playing the pure strategy i or j, respectively. So ξ and η represent primal perturbations.

Similarly, ψ and ζ stand for dual perturbations. To see that ψ_i or ζ_j indeed represents a bonus for i or j, respectively, consider the second set of equations in (3.7) with $\xi = \mathbf{0}$ for the example (2.30):

$$\begin{bmatrix} 1 & 0 & 4 \\ 0 & 2 & 4 \end{bmatrix} \begin{pmatrix} x_1' \\ x_2' \\ x_3' \end{pmatrix} + \begin{pmatrix} z_1 + \zeta_1 \\ z_2 + \zeta_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

If, say, $\zeta_2 > \zeta_1$, then one solution is $x'_1 = x'_2 = 0$ and $x'_3 = (1-\zeta_2)/4$ with $z_2 = 0$ and $z_1 = \zeta_2 - \zeta_1 > 0$, which means that only the second strategy of player 2 is optimal, so the higher perturbation ζ_2 represents a higher bonus for that strategy (as shown in the rightmost panel in Figure 3.5). Dual perturbations are in fact more general than primal perturbations, letting $\psi = A\eta$ and $\zeta = B^\top\xi$ in (3.7). Here, only special cases of these perturbations will be used, so it is useful to consider them both.

An equivalent way of writing (3.7) is

$$Dr = \mathbf{1} - D\delta \tag{3.8}$$

with the vector of variables $r \geq \mathbf{0}$, perturbation δ, and matrix D according to

$$r = (x', y', w, z)^\top, \qquad \delta = (\xi, \eta, \psi, \zeta)^\top, \qquad D = \begin{bmatrix} \mathbf{0} & A & I_m & \mathbf{0} \\ B^\top & \mathbf{0} & \mathbf{0} & I_n \end{bmatrix}. \tag{3.9}$$

The Lemke–Howson algorithm, applied to the simplified LCP (3.5), (3.6) and starting from the artificial equilibrium $(x', y') = (\mathbf{0}, \mathbf{0})$, computes with basic feasible solutions to the system $Dr = \mathbf{1}$ with bases β and inverse basis matrices D_β^{-1} using the complementary pivoting rule. For a degenerate game (where the computed equilibrium may be unstable), that rule is made unique with the lexicographic method (see Section 2.6). This method is used here as well, with a special, slightly different right hand side in (3.8). Let $k = 2(m+n)$ and $\delta = (\varepsilon, \varepsilon^2, \ldots, \varepsilon^k)^\top$. Recall that then, if ε is sufficiently small, a basis β defines a basic feasible solution to (3.8) iff the matrix $D_\beta^{-1}[\mathbf{1}, -D]$ is lexico-positive. That matrix is exactly the tableau needed for pivoting, so it is easy to compute with lexico-feasible bases using the lexico-minimum ratio test. The computation is unique since $[\mathbf{1}, -D]$ has full row rank. As before, the lexicographic method *simulates* the perturbation (with ε vanishing) rather than actually performing it.

With $\delta = (\varepsilon, \varepsilon^2, \ldots, \varepsilon^k)^\top$, the most perturbed variable is $r_1 = x_1$, where $\xi_1 = \varepsilon$. This represents essentially a primal perturbation of the first pure strategy of player 1. The other perturbations are of smaller and smaller orders of magnitude. Such a primal or dual perturbation can be achieved for *each* pure strategy by *cyclically shifting* the lexicographic order. For $1 \leq i \leq k$, the ith shift is represented by

$$\delta = (\varepsilon^{k-i+1}, \ldots, \varepsilon^k, \varepsilon, \ldots, \varepsilon^{k-i})^\top, \tag{3.10}$$

that is, $\delta_j = \varepsilon^{k-i+j}$ for $1 \leq j \leq i$ and $\delta_j = \varepsilon^{j-i}$ for $i < j \leq k$. Then the least perturbed variable (by $\delta_i = \varepsilon^k$) is r_i, and the most perturbed variable (by $\delta_{i+1} = \varepsilon$) is r_{i+1} (or r_1 if $i = k$). Call a basic solution to (3.8) that is feasible for this perturbation (3.10) *i-lexico-feasible*. This means that for each row of $D_\beta^{-1}[\mathbf{1}, -D]$, the first nonzero element is positive when inspecting the columns in the order $0, i+1, \ldots, k, 1, 2, \ldots, i$. A set of equilibria surviving these perturbations is called simply stable.

Definition 3.1. *(Wilson, 1992.) Let (A, B) be an $m \times n$ bimatrix game, $A, B > \mathbf{0}$, and $k = 2(m + n)$. 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 δ as in (3.10), there is a solution $r = (x', y', w, z)^\top \geq \mathbf{0}$ to (3.8), (3.6) so that the strategy pair (x, y) defined by (2.25) is near that set.*

Due to the perturbation, (x, y) in Def. 3.1 is only an "approximate" equilibrium. When ε vanishes, then (x, y) becomes a member of the simply stable set. Unfortunately, a simply stable set can usually not be computed by just finding an equilibrium with the Lemke–Howson algorithm started from $(\mathbf{0}, \mathbf{0})$ for each shifted lexicographic order in (3.10). The reason is that this may not always lead to the same equilibrium (and in fact such an equilibrium may not exist as Figure 3.3 shows), so that the set of these equilibria is not connected. Instead, the method invented by Wilson (1992) computes a unique *path* where on the last part of that path (representing the connected simply stable set), all points are equilibria, and all perturbations (3.10) for $i = 1, \ldots, k$ occur somewhere.

The computed path in this algorithm is piecewise composed of parts of Lemke–Howson paths. Each of the latter is here characterized by a single parameter i, for $1 \leq i \leq k$, that represents both the shift in (3.10) and the missing label. That is, the variable r_i is the least perturbed one, and it is the only variable that may be basic together with its complement (recall from (3.6) that complementary pairs of variables are x'_l, w_l and y'_j, z_j). Call such a path an *i-path*. Its vertices are characterized by the two parameters β and i. Each basis β is i-lexico-feasible, and contains only one basic variable of each complementary pair except possibly of the pair containing r_i.

For $1 \leq i < k$, a point on an i-path can be joined to an $(i + 1)$-path at an equilibrium. We describe these possibilities as they may arise in the computation. The start is from $(\mathbf{0}, \mathbf{0})$ on a 1-path until a complementary basis is reached, defining an equilibrium. Now i is increased from 1 as long as this basis stays i-lexico-feasible. In the simplest case, as when all basic variables are positive, this holds for all i up to k and the equilibrium forms by itself a simply stable set.

In general, there is some maximal $i < k$ such that the (complementary) basis is i-lexico-feasible (Wilson, 1992, p. 1060, calls this basis a *maximal i-vertex*). Then at least one basic variable is zero. Now, it is possible to let r_{i+1} or its complement enter the basis such that the resulting new basis, called a *boundary vertex*, is on an $(i + 1)$-path. This defines a certain pivoting step, called the *NP rule*, for switching from the end of an i-path to the boundary vertex which is somewhere on an $(i+1)$-path. This pivoting step is unique because of the disjoint structure of the tableau (with two parts for the two players), the fact that the old basis is not $(i + 1)$-lexico-feasible, and the definition of i-paths. Moreover, the unique leaving variable is one of the basic variables with value zero, so the pivoting step is degenerate and only changes the basis but not the feasible solution (the actual values for basic variables are always those for the unperturbed system since the perturbation is only simulated). In other words, the new boundary vertex, encoded by a new basis β and now part of an $(i+1)$-path, is still an equilibrium, that is, the respective basic solution fulfills (3.6). It happens that of the two directions that can now be taken on the $(i+1)$-path, one will destroy the equilibrium property (by increasing the value of a basic variable with value zero) but the other will not. The latter direction is taken to continue on the $(i+1)$-path.

The NP rule is conceptually the most difficult part of Wilson's algorithm. Its purpose is to define a unique adjacency of a maximal i-vertex to a boundary vertex on an $(i+1)$-path. Moreover, this adjacency relation should be symmetric so that the computed overall path is the same in either direction. That is, the NP rule should also be followed the other way around. A boundary vertex can be characterized by the following properties: it is part of an $(i+1)$-path (with $i < k$), the basis is not complementary but yet defines an equilibrium, and this property would be destroyed with the next regular pivoting step for Lemke–Howson paths (called the RP rule). In that case, it is not the RP step that is taken but the NP step (in reverse) to get back to a maximal i-vertex (Wilson, 1992, Lemma 2), which is a complementary basis.

If a complementary basis has been reached this way, the computation continues by decreasing i as long as the basis is i-lexico-feasible or until $i = 1$. At a minimal i-vertex (where the basis is not $(i-1)$-lexico-feasible), the next step of following the regular Lemke–Howson path in fact also preserves the equilibrium property (Wilson, 1992, Lemma 1).

In summary, the pivoting steps (regular or with the NP rule), and changes of the lexicographic shift i, are all performed without changing the equilibrium property, so they generate a path where all points are equilibria. Wilson remarks that the path stays in an equilibrium component, which is to be understood as a *topological* component, since the path may traverse several non-disjoint convex components. The only exception is on 1-paths which may leave or enter equilibrium components as with the ordinary Lemke–Howson algorithm. The computation terminates at a basis that is k-lexico-feasible. Since the last time the equilibrium property did not hold was on a 1-path, all perturbations $i = 1, \ldots, k$ in (3.10) have occurred in between. Therefore, that last part of the path constitutes a simply stable set. We refer to Wilson (1992) for examples and further geometric interpretations.

3.2. Perfect equilibria and the tracing procedure

Selten (1975) called an equilibrium *perfect* if it is robust against certain small mistakes of the players. We consider this concept only for normal form games. Then mistakes are represented by small positive minimum probabilities for all pure strategies, which can be chosen suitably. The resulting pair of *completely mixed* strategies converges to the equilibrium when the mistake probabilities go to zero. At the same time, the equilibrium strategies have to be best responses to these completely mixed strategies. We use this characterization of perfect equilibria (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 algorithm can also be used for computing a perfect equilibrium. The ramifications with i-paths are not necessary. It suffices to use the Lemke–Howson algorithm alone with an unshifted lexicographic order, and any missing label when starting from the artificial equilibrium.

Theorem 3.3. *Consider a bimatrix game (A, B) and, with (3.9), the LCP $Dr = \mathbf{1}$, $r \geq \mathbf{0}$, (3.6). Then the Lemke–Howson algorithm, computing with bases β such that $D_\beta^{-1}[\mathbf{1}, -D]$ is lexico-positive, terminates at a perfect equilibrium.*

Proof. Let β be the final complementary basis with basic feasible solution $r = (x', y', w, z)^\top$, and let (x, y) be the computed equilibrium obtained by (2.25). With

$$\delta(\varepsilon) = (\xi(\varepsilon), \eta(\varepsilon), \mathbf{0}, \mathbf{0})^\top = (\varepsilon, \ldots, \varepsilon^{m+n}, 0, \ldots, 0)^\top,$$

let $r(\varepsilon) = (\overline{x}(\varepsilon), \overline{y}(\varepsilon), w(\varepsilon), z(\varepsilon))^\top$ be the basic solution to $D\,r(\varepsilon) = \mathbf{1} - D\delta(\varepsilon)$ with basic variables $r_\beta(\varepsilon) = D_\beta^{-1}(\mathbf{1} - D\delta(\varepsilon))$, and zero nonbasic variables. Since $D_\beta^{-1}[\mathbf{1}, -D]$ is lexico-positive, $r(\varepsilon) \geq \mathbf{0}$ for all sufficiently small $\varepsilon \geq 0$. Similar to the equivalence of (3.7) and (3.8), we obtain

$$\begin{array}{rlrl} & A(\overline{y}(\varepsilon) + \eta(\varepsilon)) + I_m w(\varepsilon) & & = \mathbf{1}_m \\ B^\top(\overline{x}(\varepsilon) + \xi(\varepsilon)) & & + I_n z(\varepsilon) & = \mathbf{1}_n. \end{array} \tag{3.11}$$

The mixed strategies $x(\varepsilon)$ and $y(\varepsilon)$, obtained from $\overline{x}(\varepsilon) + \xi(\varepsilon)$ and $\overline{y}(\varepsilon) + \eta(\varepsilon)$ via (2.25), are completely mixed for $\varepsilon > 0$, and converge to x and y, respectively, when ε goes to zero. Furthermore, ${x'}^\top w(\varepsilon) = 0$ and ${y'}^\top z(\varepsilon) = 0$ since $w(\varepsilon)$ and w, as well as $z(\varepsilon)$ and z, have the same basic variables, and the basis is complementary, with (3.6). So x is a best response to $y(\varepsilon)$, and y is a best response to $x(\varepsilon)$. Hence, (x, y) is a perfect equilibrium according to Def. 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). It is motivated by certain shortcomings of the Lemke–Howson algorithm. First, the missing label on the computed path represents a player's pure strategy that is not a best response but is played with positive probability, whereas the other player is always in equilibrium. The computation can be interpreted by a *subsidy* for the suboptimal strategy (equal to the slack variable w_i if the missing label represents strategy i of player 1, say), which is kept while maintaining an equilibrium for both players, until either the subsidy (here w_i) or the probability for playing the strategy (here x_i) becomes zero (see also Wilson, 1992; van den Elzen, 1993). However, this is not symmetric for the two players. Furthermore, the algorithm is started from a pure strategy pair, that is, a corner of the strategy space $X \times Y$, and does not allow an arbitrary starting point. This makes the algorithm unsuitable for games with more than two players where it is sometimes useful to perform short-distance linear approximations of paths leading to an equilibrium, which requires the possibility of restarts at any position. (For "nonlinear" generalizations of the Lemke–Howson algorithm to games with more than two players see Rosenmüller, 1971; Wilson, 1971; Garcia, Lemke, and Luethi, 1973.)

The method by van den Elzen and Talman permits an arbitrary starting point, and treats both players symmetrically. Let (p, q) be an arbitrary pair of mixed strategies, that is,

$$\mathbf{1}_m^\top p = 1, \quad p \geq \mathbf{0}, \qquad \mathbf{1}_n^\top q = 1, \quad q \geq \mathbf{0}. \tag{3.12}$$

The starting point (p, q) is also called a *prior*. Consider the following constraints, with scalar variables r_0, u, v, and vectors $x, w \in \mathbb{R}^m$ and $y, z \in \mathbb{R}^n$:

$$\begin{array}{rcrrrl} & & \mathbf{1}_m^\top x & & + & r_0 = 1 \\ & & & \mathbf{1}_n^\top y + & & r_0 = 1 \\ w = \mathbf{1}_m u & & & - Ay - & (Aq) & r_0 \geq \mathbf{0} \\ z = & \mathbf{1}_n v & - B^\top x & & - (B^\top p) & r_0 \geq \mathbf{0}\,, \\ & & x\,, & y\,, & & r_0 \geq \mathbf{0}\,, \end{array} \tag{3.13}$$

and

$$x^\top w = 0, \qquad y^\top z = 0\,. \tag{3.14}$$

For $r_0 = 0$, (3.13) and (3.14) are equivalent to the familiar LCP with conditions (2.9) and (2.7), (2.8). In general, the constraints define an *augmented* LCP because of the additional column for the variable r_0 (often denoted z_0).

This augmented LCP can be solved by a complementary pivoting scheme due to Lemke (1965), which is a generalization of the Lemke–Howson algorithm. It computes with basic solutions to the system (3.13), which has $2 + m + n$ equations and $3 + 2(m + n)$ variables. The $2 + m + n$ basic variables include always u and v, and at most one variable of each complementary pair x_i, w_i and y_j, z_j for $1 \leq i \leq m$,

$1 \leq j \leq n$, so this implies (3.14). In the beginning, r_0 is also basic. In consequence, there is one complementary pair of variables that are both nonbasic, so that one of them can be chosen to enter the basis, inducing a pivoting step. The leaving variable is then either r_0, so that the algorithm terminates, or another component of x, y, w, or z, whose complement is then chosen as the next entering variable. This process continues until an equilibrium is found. (For more details see Lemke, 1965; Murty, 1988; or Cottle et al., 1992.)

Here, Lemke's algorithm can be interpreted as inducing a path in the strategy space $X \times Y$ that starts at (p, q) and ends at the computed equilibrium. The condition $0 \leq r_0 \leq 1$ will hold during the computation. Then (3.12) and (3.13) imply

$$\mathbf{1}_m^\top(x + p\,r_0) = 1, \qquad \mathbf{1}_n^\top(y + q\,r_0) = 1, \tag{3.15}$$

and $(x + p\,r_0,\ y + q\,r_0)$ is a pair of mixed strategies, which defines the path in the strategy space $X \times Y$. The vectors x and y fulfill

$$\mathbf{1}_m^\top x = 1 - r_0, \qquad \mathbf{1}_n^\top y = 1 - r_0. \tag{3.16}$$

Thus, x_i denotes how much the pure strategy i of player 1 is played *beyond* the minimum probability $p_i r_0$, and y_j denotes how much the pure strategy j of player 2 is played beyond $q_j r_0$. Furthermore, (3.13) implies

$$w = \mathbf{1}_m u - A(y + q\,r_0), \qquad z = \mathbf{1}_n v - B^\top(x + p\,r_0). \tag{3.17}$$

That is, u and v are the payoffs and w and z the corresponding slacks when playing against the mixed strategy pair $(x + p\,r_0,\ y + q\,r_0)$. Because only one variable of each complementary pair x_i, w_i and y_j, z_j is positive on the computed path, which implies (3.14), only *best responses* against the current strategy pair may have excess probabilities. In other words, the path traces equilibria of a *restricted game* where r_0 and the prior (p, q) prescribe minimum probabilities $p_i r_0$ and $q_j r_0$ for playing pure strategies. The conditions (3.16) and $x \geq \mathbf{0}$, $y \geq \mathbf{0}$ describe the player's choices in this restricted game. Initially, their freedom is zero, when $r_0 = 1$, and at the end the restricted game coincides with the full game.

Initial solutions to (3.13), (3.14) are given by $r_0 = 1$, $x = \mathbf{0}$, $y = \mathbf{0}$, and u and v sufficiently large. This does not yet define a basis since it involves $3 + m + n$ positive variables r_0, u, v, w, z. Therefore, u is chosen minimally subject to $w = \mathbf{1}_m u - Aq \geq \mathbf{0}$, so that at least one component w_i is zero and nonbasic, and replaced by x_i as a basic variable, and v is chosen minimally subject to $z = \mathbf{1}_n v - B^\top p \geq \mathbf{0}$, so that at least one component z_j is zero and made nonbasic, and the first nonbasic complementary pair is y_j, z_j.

This initial step of the algorithm amounts to finding a pure best response i of player 1 against q, and a pure best response j of player 2 against p. The prior (p, q)

can therefore be interpreted as a preconception of the players about the behavior of their opponent, against which they react initially. In general, they take the prior into account with probability r_0. The computed path shows how the players gradually adjust their behavior by using information about their opponent's actual strategy.

Given the prior (p, q) and a generic game, the pure best responses i and j against the prior are unique, with positive slacks in w and z for the remaining strategies. Letting y_j enter the basis, the value of this variable is increased and reduces the value of r_0 since (3.16) holds. We demonstrate the progress of the algorithm for the game (A, B) with

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \qquad B = \begin{bmatrix} 5 & 0 \\ 0 & 1 \end{bmatrix}. \tag{3.18}$$

Figure 3.6 shows the strategy sets X and Y of player 1 and player 2 drawn vertically and horizontally, respectively. As usual, the circled numbers 1, 2 label the pure strategies of player 1 (top and bottom row), and 3, 4 those of player 2 (left and right column), where they are best responses or have probability zero. The square represents the strategy space $X \times Y$, also subdivided into (products of) best response regions, with the three equilibria of the game marked by dots.

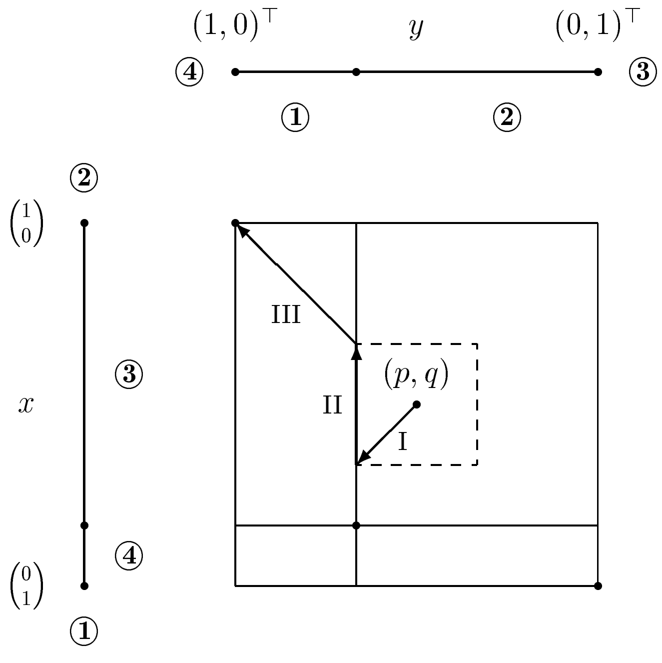

Figure 3.6. The computed path of the van den Elzen–Talman algorithm for the game (3.18) when started from the prior (p, q) where all pure strategies have equal probability.

The prior (p, q) is defined by $p_1 = p_2 = q_1 = q_2 = 1/2$. Initially, $r_0 = 1$, and the best responses to q and p are the bottom row and left column, respectively. Thus, the first basic variables besides r_0, u, v are w_1, x_2, z_2, and y_1 enters the basis. Then y_1 is increased (and with it x_2) until the slack w_1 becomes zero, which happens when $r_0 = 2/3$ and $y_1 = x_2 = 1/3$, indicated by Step I in Figure 3.6. After pivoting, y_1 is has become basic and w_1 has left the basis. At this point, both strategies of player 1 are optimal. This is maintained when in the next step, the complement x_1 of w_1 is increased. So during this part of the path (Step II), the mixed strategy $y+qr_0$ stays the same, and r_0 is also unchanged. The corresponding restricted game is shown by the dashed square. The increase of x_1 stops when x_2 becomes zero, which leaves the basis. Next, w_2 enters, so the path can leave the best response region for strategy 2 of player 1. Both x_1 and y_1 are increased, and r_0 reduced, until $r_0 = 0$ (Step III), when r_0 leaves the basis. The algorithm terminates, having reached the equilibrium (x, y) with $x = y = (1, 0)^\top$.

In general, the complementary pivoting rule determines a unique path, with lexicographic degeneracy resolution if necessary. Hence, there is no further point on the path where $r_0 = 1$ since this would reach the starting point again, and (considered backwards) allow an alternative way to start, which is not the case. So the path never leaves the strategy space. Since no basis is repeated, the algorithm eventually terminates with $r_0 = 0$. (However, it is possible that the value of r_0 sometimes increases at intermediate steps.) Furthermore, the algorithm can be used to compute a perfect equilibrium.

Theorem 3.4. *(Van den Elzen and Talman, 1991.) The complementary pivoting algorithm applied to the augmented LCP (3.13), (3.14) terminates at a perfect equilibrium if the prior (p, q) is completely mixed.*

Proof. Consider the last part of the computed path, where the variables that are allowed to have positive values are those of the final complementary basis and r_0. These $3 + m + n$ variables fulfill the $2 + n + m$ equations (with nonbasic variables set to zero) (3.15) and (3.17), that is,

$$\begin{aligned} A(y + qr_0) + I_m w \qquad &= \mathbf{1}_m u \\ B^\top(x + pr_0) \qquad\qquad + I_n z &= \mathbf{1}_n v. \end{aligned} \tag{3.19}$$

These equations define the last line segment of the path, where $r_0 > 0$ except for the endpoint determining the equilibrium (x^*, y^*). Note that u, v, x, y, w, z all vary with r_0, but the complementarity conditions (3.14) hold throughout, in particular ${x^*}^\top w = 0$ and ${y^*}^\top z = 0$. Letting $\varepsilon = r_0$ and $x(\varepsilon) = x + p\,r_0$, $y(\varepsilon) = y + q\,r_0$ in (3.19), these conditions, similar to (3.11) above, show that (x^*, y^*) is a perfect equilibrium since if $\varepsilon > 0$, then $x(\varepsilon)$ and $y(\varepsilon)$ are completely mixed strategies if p and q are. $\square$

Different priors may lead to different equilibria. For the game (3.18), Figure 3.6 shows that from the prior (p, q) with $p_1 = 1/8$, $p_2 = 7/8$ and $q_1 = q_2 = 1/2$, the algorithm reaches the bottom-right equilibrium $((0,1)^\top,(0,1)^\top)$ in a single step. Indeed, for both players the optimal response against (p,q) is their second strategy. Similar to the Lemke–Howson algorithm, this can be used to find even further equilibria, by starting the algorithm backwards from an equilibrium (with r_0 increased from zero rather than decreased from one), but with a prior leading to a different equilibrium in (3.13) (note that (p,q) serves not only as a starting point, but also as a reference throughout the computation). In Figure 3.6, this yields the unique mixed equilibrium of the game (see van den Elzen and Talman, 1991, p. 42, van den Elzen, 1993, p. 117).

Besides the symmetric treatment of the players and the freedom to choose the starting point, this algorithm is attractive because it emulates the *linear tracing procedure* of Harsanyi and Selten (1988), as shown by van den Elzen and Talman (1995). 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 $\Gamma(r_0)$. For $r_0 = 0$, this is the original game (A,B), whereas for $r_0 = 1$, it is the game with payoffs depending only on the own strategy, assuming the opponent plays according to the prior. In general, the prior is played with probability r_0, so that player 1 receives in $\Gamma(r_0)$ expected payoff $r_0 \cdot \overline{x}^\top(Aq) + (1-r_0)\cdot \overline{x}^\top(A\overline{y})$, and player 2 receives payoff $r_0 \cdot (p^\top B)\overline{y} + (1-r_0)\cdot(\overline{x}^\top B)\overline{y}$. The procedure starts with $r_0 = 1$, where $\overline{x}$ and $\overline{y}$ are the players' optimal responses to the prior. Then r_0 is decreased, changing $(\overline{x},\overline{y})$ such that it stays an equilibrium of $\Gamma(r_0)$. In this way, the players initially expect the prior and simultaneously and gradually adjust their expectations and react optimally against these revised expectations, until reaching an equilibrium of $\Gamma(0)$.

This process corresponds to the above computation, letting $x = \overline{x}(1-r_0)$ and $y = \overline{y}(1-r_0)$. Then (because it is a two-person game), player 1 and player 2 receive in $\Gamma(r_0)$ the same payoffs as in the original game against the mixed strategies $y + q\,r_0$ and $x + p\,r_0$, respectively. Thus, the paths generated by the tracing procedure and by the algorithm coincide up to projection, given by the one-to-one correspondence between $(\overline{x},\overline{y})$ and (x,y) for all $r_0 < 1$. Whereas the tracing procedure traces a pair of strategies in the full strategy space and considers convex combinations of *payoffs* with weights r_0 and $1-r_0$, the complementary pivoting algorithm generates the corresponding convex combinations of *strategies* which belong to a restricted strategy space that expands and shrinks proportionally to $1-r_0$.

We conclude with remarks on *proper* equilibria, a refinement due to Myerson (1978). A proper equilibrium is like a perfect equilibrium, except that more costly mistakes have to be made with probabilities of smaller orders of magnitude. For example, the mistake probabilities for the strategies $1,2,3,\ldots$ of player 1, whenever these are not played, are $\varepsilon^1,\varepsilon^2,\varepsilon^3,\ldots$ for the perfect equilibrium computed according to Theorem 3.3, and are therefore of decreasing orders of magnitude. However, these magnitudes have no relationship to the respective payoffs in equilibrium which indicate if a mistake is costlier than another. As topic of future research, one could try to modify Wilson's algorithm so that the lexicographic order is changed during the computation such as to find a proper equilibrium. One potential difficulty is that some suboptimal strategies have the same payoff (for example in mixed strategies in subgames of extensive games, like in the example by Wilson, 1992, p. 1048), where a strict lexicographic order no longer applies.

In contrast, the mistake probabilities in the perfect equilibrium determined by Theorem 3.4 are explicitly given. Relatively, they are the same as in the prior, and are therefore all of the same order of magnitude. The computation of proper equilibria with such an explicit representation of mistakes seems not suitable for a linear method like pivoting. Talman and Yang (1994) and Yang (1996) suggest a method based on simplicial subdivisions, which are not the topic of this paper.

4. Extensive form games

Games in extensive form are represented by a tree, with players' moves corresponding to tree edges. The standard way to find an equilibrium of such a game has been to convert it to normal form, where each combination of moves of a player is a strategy, and then applying a suitable normal-form algorithm. The problem is thereby the vast increase in the description of the game since the number of pure strategies may be *exponential* in the size of the tree. Although extensive games are convenient modeling tools, their use has been partly limited due to this reason (Lucas, 1972).

A solution to this problem was already suggested by Romanovskii (1962). This Russian paper became only known after the publication of a similar, independent result by von Stengel (1996) that extended a method by Koller and Megiddo (1992) (see also Koller, Megiddo, and von Stengel, 1994). In these approaches, probabilities for playing *sequences* of moves are characterized by certain linear equations. These, in turn, can be used for computing equilibria by techniques of linear programming. Sequences are computationally advantageous since their number is *proportional* to the tree size rather than exponential. We first discuss the exponential growth of the normal form, even if it is reduced (RNF) and if the players have a bounded number of moves at a time. Then we present the sequence form, and a corresponding LP or LCP as for the normal form. Finally, we mention methods for computing *sequential* equilibria, which are an important refinement concept for extensive form games.

4.1. Extensive form and reduced normal 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 possible *moves* of the player are assigned to the outgoing edges of the decision node. The decision nodes are partitioned into *information sets*. 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. 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. Figure 3.4 shows an example of an extensive game. 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 $\{l, r\}$.

At some decision nodes, the next move is a chance move. Chance is here treated as an additional player 0 who receives no payoff and plays according to a known *behavior strategy*. A behavior strategy of player i is given by a probability distribution on C_h for all h in H_i. A *pure strategy* is a behavior strategy where each move is picked deterministically. A pure strategy of player i can be regarded as an element of $\prod_{h \in H_i} C_h$, that is, as a tuple of moves, like $\langle L, S \rangle$ for player 1 in Figure 3.4. As in the normal form, a mixed strategy is a probability distribution on pure strategies. A behavior strategy can be considered as a special mixed strategy since it defines a probability for every pure strategy.

A *sequence* of moves of player i is the sequence of his moves (disregarding the moves of other players) on the unique path from the root to some node t of the tree, and is denoted $\sigma_i(t)$. For example, for the leftmost leaf t in Figure 3.4 this sequence is LS for player 1 and l for player 2. The empty sequence is denoted $\emptyset$. Player i has *perfect recall* (Kuhn, 1953) iff $\sigma_i(s) = \sigma_i(t)$ for any nodes $s, t \in h$ and $h \in H_i$. Then the unique sequence $\sigma_i(t)$ leading to any node t in h will be denoted σ_h. Perfect recall means that the player cannot get additional information about his position in an information set by remembering earlier moves. We assume all players have perfect recall.

Let β_i be a behavior strategy of player i. The move probabilities $\beta_i(c)$ fulfill

$$\sum_{c \in C_h} \beta_i(c) = 1, \qquad \beta_i(c) \geq 0 \qquad \text{for } h \in H_i\,, \quad c \in C_h\,. \tag{4.1}$$

The *realization probability* of a sequence σ of player i under β_i is

$$\beta_i[\sigma] = \prod_{c \text{ in } \sigma} \beta_i(c). \tag{4.2}$$

An information set h in H_i is called *relevant* under β_i iff $\beta_i[\sigma_h] > 0$, otherwise *irrelevant*.

When using a strategy as a plan of action, moves at irrelevant information sets under that strategy will not be made and can be left unspecified. In particular, this simplifies pure strategies, resulting in the *reduced normal form* or RNF (sometimes called semi-reduced normal form) of the extensive game. For example, the second information set of player 1 in Figure 3.4 is irrelevant for the pure strategies $\langle R, S\rangle$ and $\langle R, T\rangle$. In the RNF, these two strategies are identified and denoted $\langle R, *\rangle$ where $*$ stands for an unspecified move at an irrelevant information set.

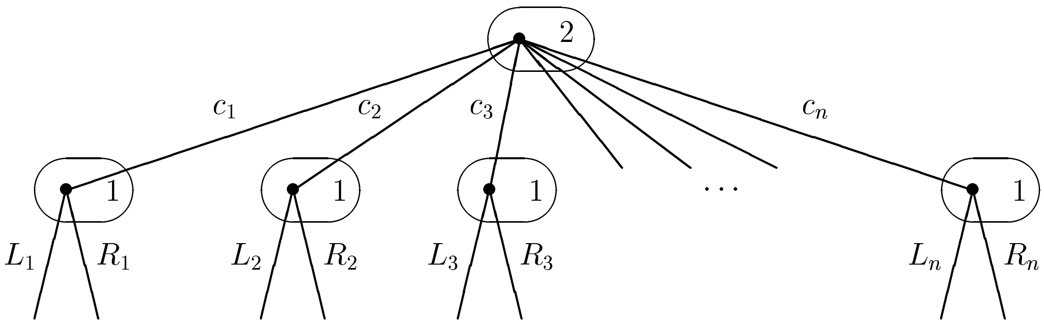

Figure 4.1. An example of a game with n possible initial moves c_j of player 2, about which player 1 is fully informed and can subsequently choose L_j or R_j, for $1 \le j \le n$. Payoffs are not shown. Player 1 has 2^n pure strategies.

The number $|\prod_{h \in H_i} C_h|$ of pure strategies is often exponential in the size of tree, unless player i has very few information sets. In the game in Figure 4.1, the tree has $3n+1$ nodes and player 1 has 2^n strategies. The RNF is the same as the full normal form. However, this game has perfect information (singleton information sets). Hence, an equilibrium is easily found by backward induction, or more generally, by recursively solving *subgames*. A subgame is a subtree that includes all information sets containing a node of the subtree, like the subtree in Figure 3.4 where player 2 moves first. Furthermore, the exponential number of strategies in the game in Figure 4.1 is due to a large number of moves of the opponent at one node.

Figure 4.2 shows a game without subgames, and where each player has only two possible moves at a time. On a path from the root to a leaf, players move alternatingly, and each player is informed about everything except the last move of the other player. At each level l (with $l = 0$ for the root) the tree has 2^l nodes. Assume that the last level L of the leaves is even (like $L = 4$ in the picture), so both players move the same number of times until the game terminates. Player 2 has $1+4+16+\cdots+4^{L/2-1}$, that is, $(2^L-1)/3$ information sets and therefore $2^{(2^L-1)/3}$ strategies in the full normal form. The number $s(L)$ of RNF strategies of player 2 can be determined recursively. For $L = 4$ as in Figure 4.2, the strategies with move l_1 are $\langle l_1, l_2, *, l_4, *\rangle$, $\langle l_1, l_2, *, r_4, *\rangle$, $\langle l_1, r_2, *, l_4, *\rangle$, and $\langle l_1, r_2, *, r_4, *\rangle$. In general, there are two information sets h with $\sigma_h = l_1$ (the first and third information set on level 3), each of which have $s(L-2)$ "subsequent" move combinations, which can be combined in any way. So there are $(s(L-2))^2$ RNF strategies with move l_1 and the same number with move r_1 (like $\langle r_1, *, l_3, *, l_5\rangle$, etc.). This gives $s(L) = 2(s(L-2))^2$ for $L \ge 2$ and $s(0) = 1$, or explicitly $s(L) = 2^{(2^{L/2}-1)}$. In terms of the number

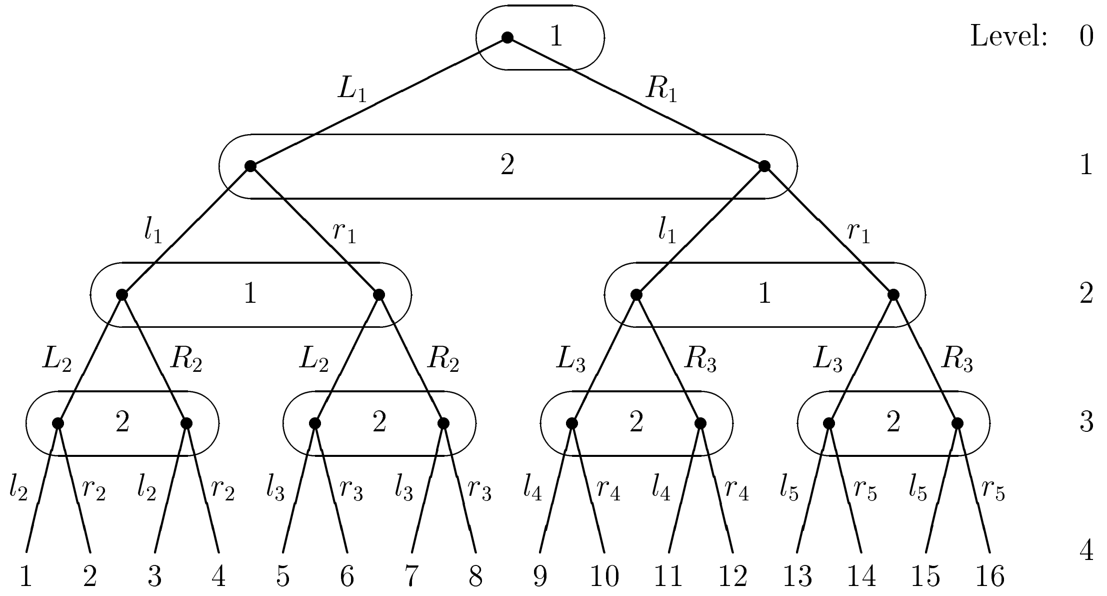

Figure 4.2. A game without subgames and two moves per decision node. For easy identification of the leaves, they have the integers 1–16 as zero-sum payoffs. Here player 1 has four and player 2 has eight RNF strategies. For this tree with n nodes, the number of RNF strategies is exponential in $\sqrt{n}$.

$n = 2^{L+1} - 1$ of nodes of the tree, this number is $2^{\sqrt{(n+1)/2}\,-1}$. This exponential growth with the square root of the size of the tree is still impractical except for small games. For $L = 10$, for example, the tree has 2047 nodes but player 2 has 2^{31} (and player 1 has 2^{16}) RNF strategies.

In this example, some information sets at the same level are distinguished by an own earlier move, so one or the other is irrelevant, which leads to a mere additive growth of the number of RNF strategies. The problem of multiplicative growth (and hence an exponential total number) arises with *parallel* information sets h and h' of a player, defined by the property $\sigma_h = \sigma_{h'}$. Such parallel information sets are due to revealed moves of the opponent. Unless these revealed moves lead to subgames which can be solved recursively, they cause computational difficulties for methods based on the normal form.

4.2. Sequence form

In order to avoid the large of size of the reduced normal form, Wilson (1972) used the Lemke–Howson algorithm with pivoting columns that are generated directly as best responses from the game tree. He observed that the computation requires few pure strategies that have positive probability, a claim made more precise by Koller and Megiddo (1996). However, it is possible to avoid pure strategies altogether by using sequences of moves instead.

Let S_i be the set of sequences (of moves) of player i. Since player i has perfect recall, any sequence σ in S_i is either the empty sequence $\emptyset$ or uniquely given by its last move c at the information set h of player i, that is, $\sigma = \sigma_h c$, so

$$S_i = \{\, \emptyset \,\} \cup \{\, \sigma_h c \mid h \in H_i,\ c \in C_h \,\}.$$

This implies that the number of sequences of player i, apart from the empty sequence, is equal to his total number of moves, that is, $|S_i| = 1 + \sum_{h \in H_i} |C_h|$. This number is linear in the size of the game tree.

The large number of mixed strategy probabilities does not arise when using behavior strategy probabilities. Let β_1 and β_2 denote behavior strategies of the two players, and let β_0 be the known behavior of the chance player. Let $a(t)$ and $b(t)$ denote the payoffs to player 1 and player 2, respectively, at a leaf t of the tree. The probability of reaching t is the product of move probabilities on the path to t. The expected payoff to player 1 is therefore

$$\sum_{\text{leaves } t} a(t)\, \beta_0[\sigma_0(t)]\, \beta_1[\sigma_1(t)]\, \beta_2[\sigma_2(t)]\,, \tag{4.3}$$

and the expected payoff to player 2 is the same expression with $b(t)$ instead of $a(t)$. However, the expected payoff is nonlinear in terms of behavior strategy probabilities $\beta_i(c)$ since the terms $\beta_i[\sigma_i(t)]$ are products by (4.2).

Therefore, we consider directly the realization probabilities $\beta_i[\sigma]$ as functions of *sequences* σ in S_i. They can also be defined for mixed strategies μ_i of player i, which choose each pure strategy π_i of player i with probability $\mu_i(\pi_i)$. Under π_i, the realization probability of σ in S_i is $\pi_i[\sigma]$, which is equal to one if π_i prescribes all the moves in σ and zero otherwise. Under μ_i, the realization probability of σ is

$$\mu_i[\sigma] = \sum_{\pi_i} \mu_i(\pi_i) \pi_i[\sigma]. \tag{4.4}$$

For player 1, this defines a map x from S_1 to $\mathbb{R}$ by $x(\sigma) = \mu_1[\sigma]$ for $\sigma \in S_1$. We call x the *realization plan* of μ_1 or a realization plan for player 1. A realization plan for player 2, similarly defined on S_2 by a mixed strategy μ_2, is denoted y. Realization plans have two important properties (Koller and Megiddo, 1992; von Stengel, 1996).

Theorem 4.1. *A realization plan x of a mixed strategy of player 1 fulfills $x(\sigma) \geq 0$ for all $\sigma \in S_1$ and*

$$\begin{aligned} x(\emptyset) &= 1, \\ \sum_{c \in C_h} x(\sigma_h c) &= x(\sigma_h) \qquad \textit{for all } h \in H_1. \end{aligned} \tag{4.5}$$

Conversely, any $x: S_1 \to \mathbb{R}$ with these properties is the realization plan of a behavior strategy of player 1, which is unique except at irrelevant information sets. A realization plan y of player 2 is characterized analogously.

Proof. Conditions (4.5) hold for the realization probabilities $x(\sigma) = \beta_1[\sigma]$ under a behavior strategy β_1 by (4.1) and (4.2) and thus for every pure strategy π_1. Hence, they also hold for the convex combinations in (4.4) with the probabilities $\mu_1(\pi_1)$.

Conversely, any x with these properties arises from the behavior strategy β_1 that makes move c at h in H_1 with probability $\beta_1(c) = x(\sigma_h c)/x(\sigma_h)$ if $x(\sigma_h) > 0$. This is necessary in order to obtain $x(\sigma) = \beta_1[\sigma]$, where (4.2) follows by induction on the length of a sequence. If $x(\sigma_h) = 0$, then h is irrelevant under β_1, and $\beta_i(c)$ for c in C_h can be defined arbitrarily so that (4.1) holds. $\square$

For the second property, two mixed strategies are called *realization equivalent* (Kuhn, 1953) if they reach any node of the tree with the same probabilities, given any strategy of the other player. We assume w.l.o.g. that all chance probabilities $\beta_0(c)$ are positive.

Theorem 4.2. *Two mixed strategies μ_i and μ_i' of player i are realization equivalent iff they have the same realization plan, that is, iff $\mu_i[\sigma] = \mu_i'[\sigma]$ for all $\sigma \in S_i$.*

Proof. Let $i = 1$, say, and let μ_2 be a completely mixed strategy of player 2. Then any node t of the game tree is reached with probability $\beta_0[\sigma_0(t)]\,\mu_1[\sigma_1(t)]\,\mu_2[\sigma_2(t)]$. This expression is the same for μ_1 and μ_1' iff $\mu_1[\sigma] = \mu_1'[\sigma]$ for all $\sigma \in S_1$, since the terms $\beta_0[\sigma_0(t)]$ and $\mu_2[\sigma_2(t)]$ are positive. $\square$

These two theorems imply the well-known result by Kuhn (1953) that behavior strategies are strategically as expressive as mixed strategies.

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

Theorem 4.1 characterizes realization plans by nonnegativity and the equations (4.5). A realization plan describes a behavior strategy uniquely except for the moves at irrelevant information sets. In particular, the realization plan of a *pure* strategy (that is, a realization plan with values zero or one) is as specific as the pure strategy in the RNF.

A realization plan represents all the relevant strategic information of a mixed strategy by Theorem 4.2. This compact information is obtained with the linear map in (4.4). This map assigns to any mixed strategy μ_i, regarded as a tuple of mixed strategy probabilities $\mu_i(\pi_i)$, its realization plan, regarded as a tuple of realization probabilities $\mu_i[\sigma]$ for σ in S_i. The simplex of mixed strategies is thereby mapped to the polytope of realization plans defined by the linear constraints in Theorem 4.1.

The vertices of this polytope are the realization plans of pure strategies. We have seen that the number of these vertices may be exponential. The number of defining inequalities and the dimension of the polytope, however, is linear in the tree size. For player i, this dimension is the number $|S_i|$ of variables minus the number $1+|H_i|$ of equations (4.5) (which are linearly independent), so it is $\sum_{h\in H_i}(|C_h|-1)$.

We consider realization plans as vectors in $x \in \mathbb{R}^{|S_1|}$ and $y \in \mathbb{R}^{|S_2|}$, that is, $x = (x_\sigma)_{\sigma\in S_1}$ where $x_\sigma = x(\sigma)$, and similarly $y = (y_\tau)_{\tau\in S_2}$. The linear constraints in Theorem 4.1 are denoted by

$$Ex = e, \quad x \geq \mathbf{0} \qquad \text{and} \qquad Fy = f, \quad y \geq \mathbf{0}, \tag{4.6}$$

using the *constraint* matrices E and F and vectors e and f. The matrix E and right hand side e have $1+|H_1|$ rows, and E has $|S_1|$ columns. The first row denotes the equation $x(\emptyset) = 1$ in (4.5). The other rows for $h \in H_1$ are the equations $-x(\sigma_h) + \sum_{c\in C_h} x(\sigma_h c) = 0$. For the game in Figure 4.2, listing the sequences of player 1 as $\emptyset$, L_1, R_1, L_1L_2, L_1R_2, R_1L_3, R_1R_3,

$$E = \begin{pmatrix} 1 & & & & & & \\ -1 & 1 & 1 & & & & \\ & -1 & & 1 & 1 & & \\ & & -1 & & & 1 & 1 \end{pmatrix}, \qquad e = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{4.7}$$

Similarly, F and f have $1+|H_2|$ rows. In Figure 4.2,

$$F = \begin{pmatrix} 1 & & & & & & & & & & \\ -1 & 1 & 1 & & & & & & & & \\ & -1 & & 1 & 1 & & & & & & \\ & & -1 & & & 1 & 1 & & & & \\ & -1 & & & & & & 1 & 1 & & \\ & & -1 & & & & & & & 1 & 1 \end{pmatrix}, \qquad f = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Each sequence appears exactly once on the left hand side of the equations (4.5), accounting for the entry 1 in each column of E and F.

Define the *sequence form payoff* matrices A and B, each of dimension $|S_1| \times |S_2|$, as follows. For $\sigma \in S_1$ and $\tau \in S_2$, let the matrix entry $a_{\sigma\tau}$ of A be defined by

$$a_{\sigma\tau} = \sum_{\substack{\text{leaves } t\,: \\ \sigma_1(t)=\sigma,\ \sigma_2(t)=\tau}} a(t)\, \beta_0[\sigma_0(t)]\,. \tag{4.8}$$

The matrix entry of B is this term with b instead of a. These two matrices are *sparse*, since the matrix entry for a pair σ, τ of sequences is zero (the empty sum) whenever these sequences do not lead to a leaf. If they do, the matrix entry is the payoff at the leaf (or leaves, weighted with chance probabilities of reaching the leaves, if there are chance moves). Then by (4.3) and (4.6), the expected payoffs to player 1 and player 2 are $x^\top Ay$ and $x^\top By$, respectively, which is just another way of writing the weighted sum over all leaves. Figure 4.3 shows an example of a sequence form payoff matrix. The constraint and payoff matrices define the *sequence form* of the game.

	$\emptyset$	l_1	r_1	l_1l_2	l_1r_2	r_1l_3	r_1r_3	l_1l_4	l_1r_4	r_1l_5	r_1r_5
$\emptyset$											
L_1											
R_1											
L_1L_2				1	2	5	6				
L_1R_2				3	4	7	8				
R_1L_3								9	10	13	14
R_1R_3								11	12	15	16

Figure 4.3. Sequence form payoff matrix for the game in Figure 4.2.

4.3. Computing equilibria

The computation of equilibria with the sequence form uses the same approach as presented for the normal form in Sections 2.2 and 2.3. Consider a fixed realization plan y of player 2. An optimal realization plan x of player 1 maximizes his expected payoff $x^\top(Ay)$, subject to $Ex = e$, $x \geq \mathbf{0}$. This LP has a dual LP with a vector u of unconstrained variables whose dimension is $1 + |H_1|$, the number of rows of E. This dual LP, analogous to (2.4), is to

$$\begin{aligned} &\text{minimize} && e^\top u \\ &\text{subject to} && E^\top u \geq Ay. \end{aligned} \tag{4.9}$$

If the game is *zero-sum*, then player 2 is interested in minimizing the optimal payoff to player 1, which is the value of this dual LP. So a minmax realization plan y of player 2 solves the LP with variables u, y which is analogous to (2.5),

$$\begin{aligned} \text{minimize} \quad & e^\top u \\ \text{subject to} \quad & \qquad\quad Fy = f, \\ & E^\top u - Ay \geq \mathbf{0}, \\ & \qquad\qquad\; y \geq \mathbf{0}. \end{aligned} \tag{4.10}$$

It is easy to see that the dual of this LP, analogous to (2.6), determines a maxmin realization plan of player 1. The number of nonzero entries in the sparse matrices E, F, A, and the number of variables, is linear in the size of the extensive game. Hence, we have shown the following.

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 (4.10) with sparse payoff matrix A in (4.8) and constraint matrices E and F in (4.6) defined by Theorem 4.1. The size of this LP is linear in the size of the game tree.*

Romanovskii (1962) constructed the sequence form payoff matrix with entries $a_{\sigma\tau}$ in (4.8). As here, rows and columns are played with variables x_σ and y_τ constrained by (4.6). He used the construction of Charnes (1953) for solving such a *constrained matrix game*. As mentioned, his result was overlooked.

Koller and Megiddo (1992) defined the constraints (4.5) for playing sequences for a player with perfect recall. For the other player, they still considered pure strategies since they used a single payoff variable instead of a dual vector u as in (4.9), and since they did not construct a symmetric payoff matrix. They defined an LP with a linear number of variables x_σ but possibly exponentially many inequalities. However, these can be evaluated as needed by finding a best response of the opponent, which can be done quickly by backward induction as shown by Wilson (1972). This solves efficiently the "separation problem" for the ellipsoid method for linear programming, which therefore runs in polynomial time. Apart from Romanovskii's result, this was the first polynomial-time algorithm for solving two-person zero-sum games in extensive form.

For non-zero-sum games, we derive analogous to Section 2.3 an LCP. For player 2, the realization plan y is a best response to x iff it maximizes $(x^\top B)y$ subject to $Fy = f$, $y \geq \mathbf{0}$. Its dual LP has the vector v of variables and says: minimize $f^\top v$ subject to $F^\top v \geq B^\top x$. The optimality of this primal-dual pair of LPs, as well as that for player 1, is characterized by complementarity conditions analogous to (2.8) and (2.7). We obtain the result corresponding to Theorem 2.4.

Theorem 4.5. *Consider the two-person extensive game with sequence form payoff matrices A, B and constraint matrices E, F in (4.6). Then the pair (x, y) of realization plans defines an equilibrium iff there are vectors u, v such that*

$$\begin{array}{rrr} & Ex & = e \\ & & Fy = f \\ E^\top u & & -\,Ay \geq \mathbf{0} \\ & F^\top v - B^\top x & \geq \mathbf{0} \\ & x\,, & y \geq \mathbf{0} \end{array} \tag{4.11}$$

and

$$x^\top(E^\top u - Ay) = 0\,, \qquad y^\top(F^\top v - B^\top x) = 0\,.$$

The size of this LCP with variables u, v, x, y is linear in the size of the game tree.

Compared to the LCP for the normal form, the sequence form gives an LCP of small size. So far, the standard Lemke–Howson algorithm has not been applied to this LCP since the "missing label" corresponds to a pure strategy. Here, a pure strategy is usually a realization plan that plays several sequences with positive probability. Using the Lemke–Howson algorithm for the sequence form requires labels for sequences, but would probably have to drop and pick up several labels at a time. It is not clear how this can be done.

Instead, it is possible to use Lemke's algorithm using a *prior* (p, q) of realization plans analogous to the algorithm by van den Elzen and Talman described in Section 3.2. By assumption, $Ep = e$, $p \geq \mathbf{0}$, $Fq = f$, $q \geq \mathbf{0}$. Analogous to (3.13), which adds an extra column to (2.9), we add a column to (4.11), and compute with solutions to the system

$$\begin{array}{rcccl} & Ex & + & & e\,r_0 = e \\ & & & Fy\,+ & f\,r_0 = f \\ w = E^\top u & & & -\,Ay\,- & (Aq)r_0 \geq \mathbf{0} \\ z = & F^\top v - B^\top x & & & -\,(B^\top p)r_0 \geq \mathbf{0}\,, \\ & x\,, & & y\,, & r_0 \geq \mathbf{0} \end{array} \tag{4.12}$$

and

$$x^\top w = 0, \qquad y^\top z = 0\,.$$

Koller, Megiddo, and von Stengel (1996) suggested to use Lemke's complementary pivoting algorithm for such a system, although not with this particular form of the column for the extra variable r_0. Using such a generally defined column requires a rather technical proof that Lemke's algorithm terminates.

With the constraints (4.12), one can use the same interpretation as for the normal form algorithm in Section 3.2. That is, the algorithm starts with $r_0 = 1$. Throughout the computation, $(x + p\,r_0,\; y + q\,r_0)$ is a pair of realization plans which never leaves the strategy space. Therefore, the algorithm terminates with $r_0 = 0$. If all sequences (and hence all moves and pure strategies) have positive probability under (p, q), then the computed equilibrium is *normal form perfect.* Furthermore, the computation mimicks the linear tracing procedure of Harsanyi and Selten applied to the normal form of the game. These results and the close relationship of the sequence form with the normal form are shown by von Stengel, van den Elzen, and Talman (1996).

The normal form of an extensive game is typically degenerate, as Figure 3.4 shows. The reason is that optimal strategies may avoid entire branches of the tree, so the number of used strategies (or moves) is usually not the same for both players. Therefore, degeneracy arises also in the sequence form, and must be resolved by lexicographic pivoting methods. An example is again the game in Figure 3.4 because its sequence form is equivalent to the RNF. This is always the case if the game has no parallel information sets. Then each column of E and F has at most one entry -1, like E in (4.7), and the realization probabilities for maximal sequences sum to one. In other words, the maximal sequences of a player without parallel information sets define unique RNF pure strategies.

Further examples and details on the sequence form can be found in the cited papers. For example, the components of the dual vectors u and v in (4.10) or (4.11) can be interpreted as partial payoffs for optimal moves at information sets. The sequence form is naturally defined for more than two players, although the pivoting algorithms no longer apply.

Koller and Megiddo (1992) have shown that solving a zero-sum two-person game in extensive form with *imperfect recall* is NP-complete. Hence, no efficient algorithm for this problem is likely to exist. Koller and Megiddo (1996) applied sparse basis representations to extensive two-person games for enumerating *all* equilibria of the game in time that is exponential in the tree size. This follows from Theorem 4.5 when using the sequence form, but remarkably works also for games with imperfect recall (the additional effort merely doubles the exponent in the running time).

Finally, we mention two unrelated papers that consider the problem of finding *sequential equilibria*. This is a refinement for extensive form games due to Kreps and Wilson (1982), which considers equilibria that have to be consistent with suitable *beliefs* about one's position in an information set. Some algorithmic aspects of this concept are mentioned in the paper itself. Azhar, McLennan, and Reif (1993) formulated LP conditions for determining if a belief system is consistent. A similar characterization is given by Perea y Monsuwé, Jansen, and Peters (1996).

5. Computational issues

How long does it take to find an equilibrium of a bimatrix game? The Lemke–Howson algorithm is known to take exponential time for some specifically constructed, even zero-sum, games. This is similar to certain LPs where the Simplex algorithm has exponential running time. However, the Simplex algorithm has been observed as very efficient for most practical problems. The same is said for complementary pivoting methods, although the solved LCPs are much smaller due to the overriding problems of numerical stability (Tomlin, 1978; Cottle et al., 1992). Interior point methods that are provably polynomial as they are known for linear programming are not yet known for LCPs arising from games.

The computational complexity of finding one equilibrium is unclear. The decision problem if there is an equilibrium is trivial by Nash's theorem, but finding a particular equilibrium requires obviously more. Megiddo and Papadimitriou (1989) defined a new complexity class of such problems where solutions exist but must be found constructively. Papadimitriou (1994) showed in a similar complexity class the computational *equivalence* of related results like Nash's theorem, Sperner's lemma, or the existence of fixed points of a continuous mapping on a simplex. The approximation of such fixed points takes worst-case exponential time in the dimension and the number of digits of accuracy if the mapping is evaluated as an oracle (Hirsch, Papadimitriou, and Vavasis, 1989). All known fixed-point approximations are such oracle algorithms, but a more efficient method might well "look into" the function and use its specific representation, like the function used in the proof of Nash's theorem (1951).

Gilboa and Zemel (1989) showed that finding an equilibrium of a bimatrix game with maximum payoff sum is NP-hard, so this is a problem where no efficient solution is likely to exist. The same holds for other problems that amount essentially to examining all equilibria, like finding an equilibrium with maximum support. (Note that Theorem 2.15, for example, requires to inspect the cliques of a graph. In general, finding a maximum-sized clique is NP-hard.) Koller, Megiddo, and von Stengel (1994) cite miscellaneous papers connecting theoretical computer science and game theory.

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 (see its overview in McKelvey and McLennan, 1996, or its World Wide Web site at http://www.hss.caltech.edu/~gambit/Gambit.html). This is a program for building and solving extensive games interactively or with the help of a command language. For the automatic generation of large extensive games with regular structure like in Figure 4.2, Koller and Pfeffer (1997) developed a PROLOG-based language GALA. One application was a simplified version of Poker. From simply defined rules, GALA generated a game tree of 50,000 nodes and a corresponding LP according to Theorem 2.4, which was solved by a commercial LP solver in minutes. These program systems are under development to become efficient and numerically stable implementations of algorithms 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. Mathematical 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 Computational Geometry 8, 295–313.

S. Azhar, A. McLennan, and J. H. Reif (1993), Computation of equilibria in noncooperative games. Mimeo, Duke University and University of Minnesota.

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

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.

G. B. Dantzig (1991), Linear Programming. In: History of Mathematical Programming – A Collection of Personal Reminiscences, eds. J. K. Lenstra, A. H. G. Rinnooy Kan, A. Schrijver. CWI and North-Holland, Amsterdam, 19–31.

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

B. C. Eaves (1971), The linear complementarity problem. Management Science 17, 612–634.

D. Gale, H. W. Kuhn, and A. W. Tucker (1950), On symmetric games. In: Contributions to the Theory of Games I, eds. H. W. Kuhn and A. W. Tucker, Annals of Mathematics Studies 24, Princeton Univ. Press, Princeton, 81–87.

C. B. Garcia, C. E. Lemke, and H. Luethi (1973), Simplicial approximation of an equilibrium point for non-cooperative N-person games. In: Mathematical Programming, eds. T. C. Hu and S. M. Robinson, Academic Press, New York, pp. 227–260.

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.

M. Hirsch, C. H. Papadimitriou, and S. Vavasis (1989), Exponential lower bounds for finding Brouwer fixed points. Journal of Complexity 5, 379–416.

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.

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.

D. M. Kreps and R. Wilson (1982), Sequential equilibria. Econometrica 50, 863–894.

I. Krohn, S. Moltzahn, J. Rosenmüller, P. Sudhölter, and H.-M. Wallmeier (1991), Implementing the modified LH algorithm. Applied Mathematics 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, 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.

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 of bimatrix games. Journal of the Society for Industrial and Applied Mathematics 12, 778–780.

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 (1996), Generic 4×4 two person games have at most 15 Nash equilibria. University of Minnesota and University of Bristol.

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

N. Megiddo and C. H. Papadimitriou (1989), A note on total functions, existence theorems, and computational complexity. IBM Research Report RJ 7091.

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.

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.

R. B. Myerson (1978), Refinements of the Nash equilibrium concept. International Journal of Game Theory 7, 73–80.

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.

A. Perea y Monsuwé, M. Jansen, and H. Peters (1996), Characterization of fully consistent assessments in extensive form games. Discussion paper, University of Limburg, Maastricht.

T. Quint and M. Shubik (1994), On the number of Nash equilibria in a bimatrix game. Dept. of Operations Research, Yale University, New Haven.

I. V. Romanovskii (1962), Reduction of a game with complete memory to a matrix game. Soviet Mathematics 3, 678–681. (English translation of Doklady Akademii Nauk SSSR 144, 62–64.)

J. Rosenmüller (1971), On a generalization of the Lemke–Howson algorithm to noncooperative N-person games. SIAM Journal on Applied Mathematics 21, 73–79.

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.

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, 367–377.

A. J. J. Talman and Z. Yang (1994), A simplicial algorithm for computing proper equilibria of finite games. Discussion paper No. 9418, Center for Economic Research, Tilburg.

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

M. J. Todd (1978), Bimatrix games – an addendum. Mathematical 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.

A. van den Elzen (1993), Adjustment Processes for Exchange Economies and Noncooperative Games. Lecture Notes in Economics and Mathematical 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.

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

B. von Stengel (1997), New maximal numbers of equilibria in bimatrix games. Technical Report 264, Dept. of Computer Science, ETH Zürich.

B. von Stengel, A. H. van den Elzen, and A. J. J. Talman (1996), Tracing equilibria in extensive games by complementary pivoting. Discussion paper No. 9686, Center for Economic Research, Tilburg.

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

R. Wilson (1971), Computing equilibria of N-person games. SIAM Journal on Applied Mathematics 21, 80–87.

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 Mathematical Economics, eds. O. Moeschlin and D. Pallaschke, North-Holland, Amsterdam, 137–148.

Z. Yang (1996), Simplicial Fixed Point Algorithms and Applications. CentER dissertation, Tilburg University.

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