Equilibrium Computation for Extensive Games

Wan Huang

A thesis submitted for the degree of

Doctor of Philosophy

Department of Mathematics
London School of Economics
and Political Science

January 27, 2011

Declaration

I certify that the thesis I have presented for examination for the MPhil/PhD degree of the London School of Economics and Political Science is based on joint work with Bernhard von Stengel.

I warrant that this authorization does not, to the best of my belief, infringe the rights of any third party.

Abstract

This thesis studies equilibrium computation algorithms for extensive games. We focus on the enumeration of Nash equilibria and on the computation of an extensive form correlated equilibrium.

The contribution of this thesis consists of two parts. First, we study an algorithm for enumerating all Nash equilibria of a two-player extensive game. This algorithm is based on the sequence form description for Nash equilibria of extensive games (von Stengel 1996). We develop a systematic way of eliminating redundancy in this system, and prove that all the equilibria are represented by vertices of a pair of polyhedra. Then we apply the reverse search vertex enumeration algorithm by Avis (2000) to this pair of polyhedra to enumerate all the vertices. We use a label system to verify pairs of vertices that represent Nash equilibria.

Second, we present a polynomial time algorithm for computing an extensive form correlated equilibrium (EFCE) of a multi-player game with chance moves. To achieve this, we first characterize the EFCE as product distributions that satisfy a set of incentive constraints. We then provide a constructive proof of the existence of EFCE. Based on this proof, we show that an EFCE of a multi-player game with chance moves can be computed in polynomial time.

Acknowledgments

First of all, I am heartily thankful to my supervisor, Professor Bernhard von Stengel, who provided encouragement, guidance and support. Professor Bernhard von Stengel contributed countless hours in improving the writing and proof-reading of my thesis paper, and discussing the topics. This dissertation would not have been possible without his direct supervision.

Also I would like to give special thanks to LSE who granted me the scholarship. And I offer my regards and blessings to all of those in the school who supported me during the preparation and completion of my study. In particular I want to give special thanks to the administrators in the Mathematics Department who provided logistic assistance.

Last but not the least, I owe my deepest gratitude to my family who supported me all these years financially and emotionally.

Contents

CHAPTER 1

Introduction

This thesis discusses two equilibrium computation problems of extensive games with perfect recall: the enumeration of Nash equilibria, and the computation of one extensive form correlated equilibrium (EFCE). The main results are accordingly explained in two chapters. The first is presented in Chapter 2 which deals with the enumeration of Nash equilibria. In it we recall the linear complementarity problem that characterizes the set of Nash equilibria for the sequence form (von Stengel 1996). We identify the redundancy in the system and obtain a reduced system by removing the redundant variables and constraints. We investigate several algorithms computing Nash equilibria for bimatrix games, and extend these algorithms to extensive games. Chapter 3 contributes to the study of extensive form correlated equilibria (EFCE). We give one characterization of the set of EFCE, an existence proof, and describe the computation of one of the EFCE for multi-player extensive games.

1.1 Contribution of this thesis

In Chapter 2, we develop an algorithm of computing all equilibria for two-player *extensive games*. Analogously to the equilibrium computation algorithm by Avis et al. (2010) for *bimatrix games* that computes all the vertices of a pair of *best response polytopes*, this algorithm computes all the vertices of a pair of polyhedra. Unlike the algorithm for bimatrix games that uses the strategic form to construct the best response polytopes, the algorithm for extensive games developed in this thesis constructs the pair of polyhedra based on the sequence form (von Stengel 1996), because the sequence form is of smaller dimension than the strategic form.

Unfortunately, using the lower-dimensional sequence form no longer allows us to transform the polyhedra to polytopes, which are useful for vertex enumeration, because the sequence form has several linear constraints and therefore several dual variables which cannot be eliminated the way that the scalar payoff variable of a player in the strategic form is eliminated.

On the other hand, it is possible to reduce the dimension of the polyhedra obtained for an extensive game. The dual variables can be interpreted as partial payoffs at information sets. Because of multiple dual variables, there may be some pairs of points that represent the same extreme equilibria. These pairs of points only differ in their value of the multiple dual variables. Since our goal is finding Nash equilibria, we are not particularly interested in these variables, so it is useful to eliminate as many of these variables as possible. The goal is to have a dimension, in terms of number of variables, that is never larger than the dimension of the strategic form, and this is possible.

We describe a systematic way of eliminating variables based on the sequences in the game tree. The resulting system has lower dimension than the original one. Unfortunately, writing all this down is less elegant than the original sequence form with its constraints (see Theorem 2.7).

Multiple dual variables also cause the problem of considering points on extreme rays as part of extreme equilibria. We prove that all Nash equilibria are represented by pairs of vertices. Hence, points on extreme rays cannot represent Nash equilibria. We do this by showing that the dual system for the original unreduced sequence form has only bounded solutions in equilibrium, which then carries over to the dual variables of the reduced sequence form obtained by eliminating primal variables with the help of the given equations. The argument for this (see Lemma 2.17) is general and possibly of independent interest.

Chapter 3 explores the computation of an EFCE for multi-player games. Starting with games without chance moves, we develop a system of consistency and incentive constraints to describe the set of EFCE for multi-player games. We prove the existence and present a polynomial time algorithm for computing an EFCE. Then we extend the proof and the algorithm to games with chance moves.

For two-player extensive games without chance moves, von Stengel and Forges (2008) provide a *compact* description, which is a system of constraints of the realization plans. This description, however, does not apply to multi-player games or games with chance moves. In such games, the consistency and incentive constraints are necessary but not sufficient conditions. For this reason, we use constraints of strategy profiles instead of realization plans in our system to describe the set of EFCE for multi-player games.

The existence proof and the algorithm for computing an EFCE is analogous to that for a strategic form correlated equilibrium (CE) by Papadimitriou and Roughgarden (2008). The existence proof for both CE and EFCE exploit the duality theorem of linear programming. The proof for EFCE involves some tedious but straightforward proof of properties which are derived mainly from the tree structure and by using induction.

Papadimitriou and Roughgarden (2008) claim that a correlated equilibrium of a *succinctly-representable* game can be computed in polynomial time. Their algorithm is based on a polynomial time algorithm called the *ellipsoid method*, which applies to linear strict inequalities (LSI). In a paper recently uploaded to arXiv, Stein et al. (2010) pointed out that this algorithm can fail even for small games, due to numerical precision issues. Papadimitriou (2010) acknowledged the need to update his algorithm in an online blog and expressed confidence that this issue could be overcome without dramatic changes. Since then and possibly in response to this comment, Stein et al. have withdrawn their paper. Jiang and Leyten-Brown (2010) recently presented a variant of Papadimitriou and Roughgarden's algorithm, and showed that the numerical precision issue could be overcome by their algorithm. We elaborate on this discussion in Section 3.3.

Chapter 3 is joint work with Bernhard von Stengel. A version of this chapter has been presented in WINE 2008 and published in Lecture Notes in Computer Science (Huang and von Stengel 2008).

1.2 Extensive games

In this dissertation, the words *extensive game* always refer to a finite game in extensive form. Extensive games are a standard description of strategic situations in game theory. A game of this kind is a "dynamic" description of an interactive situation, compared with the strategic form that gives a "static" description.

We first state some standard vocabulary for extensive games. In an extensive game, the players act sequentially. Some moves can made by *chance* (or nature). An extensive game is represented as a *game tree*. The basic structure is a directed tree. Each nonterminal node (called a *decision node*) represents a possible state of play of the game as it is played. The game begins at a unique initial node, which is the root of the tree, and goes through the tree along a path determined by the players' moves at each node, until a terminal node (which is a leaf of the tree) is reached, where the game ends and payoffs are assigned to all players. Each nonterminal node belongs to either a player or to chance. For the node of a player, that player chooses among the possible moves at that node; for a chance node, chance can be considered as an additional player that receives no payoff and always plays according to a strategy with probabilities that are given in the description of the game.

Every nonterminal node has one or several branches, which are edges leading from this nonterminal node to other nodes. For the nodes that belong to players, each branch shows a possible move for the player at the state that the nonterminal node represents. If the edge leads to another nonterminal node, the player choosing that possible move goes to another state of the game. Otherwise, when the edge leads to a terminal node, the game is terminated and all players get their payoffs. Branches of a chance node are chance moves. They are assigned the *chance probabilities* that are nonnegative and sum up to 1.

Every terminal node is a possible outcome of the game. The sequence of edges of a path from the root to a terminal node of the tree specifies a *play* of the game.

Games where the players have imperfect information about the game state are modelled with in *information sets*, due to Kuhn (1953). The set of information sets is a partition of the decision nodes. All nodes in one information set belong to the same player and have the same moves. When the player reaches one node in this information set, he only knows the information set but not the particular node he reaches. For all nodes that are in the same information set, they have the same number of branches, representing that the play has the same set of choices at all these states. When a game has an information set with more than one node, we say that this game has imperfect information. Otherwise, if every information set of a game is a singleton set, this game has perfect information, and every player knows exactly what has taken place earlier in the game. In this dissertation, an information set is indicated by a dotted line connecting all nodes in that set.

In this dissertation, we focus on extensive games for which the players have *perfect recall*. When a player has perfect recall, at any time the player knows what moves she has done before. In other words, all the nodes of an information set of this player have the same earlier own moves.

Following the description of the extensive form by Myerson (1991), an extensive game is formally a rooted tree G, together with functions that assigns labels to every node and branch, satisfying the following conditions:

1. Each nonterminal node has a *player label* that is in the set $P = \{0,\dots,n\}$, where 0 is assigned to the chance node. The nonterminal nodes are the decision nodes.

2. Each nonterminal node, except for the chance nodes, has a second label that specifies the information set. We denote the set of the information set labels by H. Nodes with the same information set label belong to the same information set.

3. Each branch of a nonterminal non-chance node has a *move label*. The set of move labels that are assigned to branches of nodes belonging to the information set h is denoted C_h. Nodes belonging to the same information set h have the same number of branches with the same set of move labels.

4. Each branch of a chance nodes is assigned a chance probability. For each chance node, the chance probabilities are nonnegative and sum up to 1.

5. The payoff function a assigns a vector $a(s) = (a^1(s),\dots,a^n(s))$ with real numbers as components to every terminal node s.

To illustrate these concepts, consider the two extensive games in the following examples. The first is an extensive game without chance moves, the second is a game with chance moves.

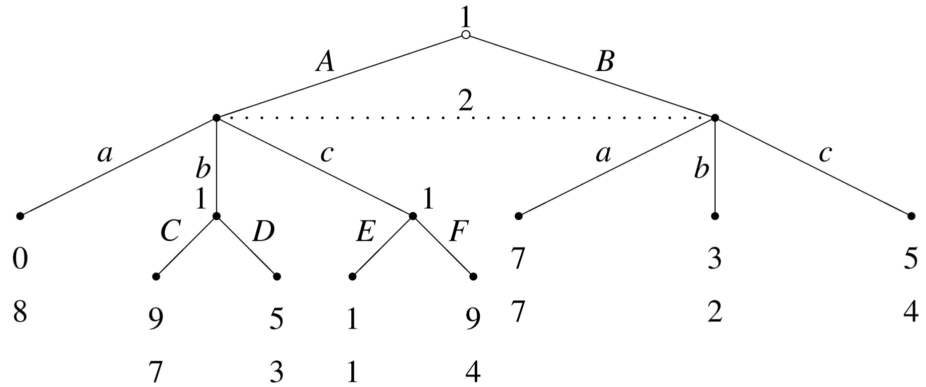


Figure 1.1: An extensive game without chance moves. At a leaf, the top payoff is to player 1 and the bottom payoff is to player 2.

Example 1.1 (An extensive game without chance moves) In Figure 1.1, moves are marked by upper-case letters for player 1 and by lower-case letters for player 2. The three information sets of player 1 have move sets $\{A,B\}$, $\{C,D\}$ and $\{E,F\}$, and the information set of player 2 has move set $\{a,b,c\}$. A play of the game is a sequence of moves that leads to a leaf of the tree. For example, player 1 chooses A, player 2 chooses b, and player 1 chooses C, after which the game terminates with payoffs 9 for player 1 and 7 for player 2. The move b of player 2 is the same no matter whether player 1 chooses A or B (but the resulting plays are of course different). Player 2 does not know the game state in her information set.

Example 1.2 (An extensive game with chance moves) In Figure 1.2, the information set of player 1 has move set $\{L,R\}$, and the information sets of player 2 have move sets $\{a,b\}$ and $\{c,d\}$. The game is initially played by the chance move.

An extensive game can be represented by the "strategic form" or the "sequence form". We explain and compare the two representations.

For the strategic form representation, a pure *strategy* of a player specifies one move for each information set of this player. The set of all pure strategies of player i

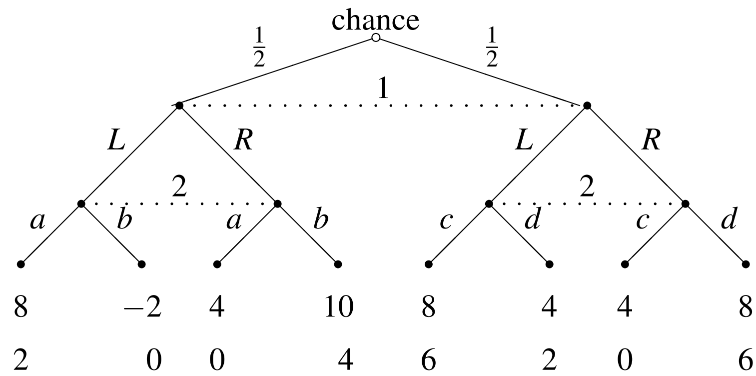


Figure 1.2: An extensive game with chance moves.

is denoted S_i, and $S = \prod_{i \in P} S_i$ is the set of all *strategy profiles*. Thus a strategy profile specifies one pure strategy for each player. For Example 1.1, the set of all strategies for player 1 is

$$S_1 = \{(ACE),(ACF),(ADE),(ADF),(BCE),(BCF),(BDE),(BDF)\},$$

where (ACE) etc. stands for (A,C,E) for brevity.

The description of these strategies can be *reduced*. Consider in the extensive game of Example 1.1 the pure strategies (BCE), (BDE), (BCF) and (BDF). The initial move B of player 1 makes the subsequent choice of C or D, and E or F irrelevant since these two information sets of player 1 cannot be reached after move B. Therefore these four strategies can be regarded as identical. For the two unreachable information sets, we let $*$ denote an arbitrary move. Thus these strategies are written as $(B{*}{*})$. This is called a *reduced strategy* of player 1. Generally, a (pure) reduced strategy of a player specifies one move for each information set of this player except for the information sets that are unreachable due to an own earlier move. The set of all reduced strategies of player i is denoted S_i^*. The set of reduced strategies for player 1 is

$$S_1^* = \{(ACE),(ACF),(ADE),(ADF),(B{*}{*})\}.$$

In this example, player 1 has two *parallel* information sets that have the sets of moves $\{C,D\}$ and $\{E,F\}$. These information sets are not distinguished by own earlier moves and arise because player 1 receives information about an earlier move by player 2. Combinations of moves at parallel information sets cannot be reduced, because by definition two information sets are parallel if they are preceded by the same sequence of own earlier moves of the player.

The strategic form (or normal form) of an extensive game is given by tabulating all pure strategies of the players and recording the resulting expected payoffs. The reduced strategic form is given by tabulating all reduced strategies of the players and the expected payoffs. The strategic form of the extensive game in Example 1.2 is

1 \ 2	L	R
ac	8, 4	4, 0
ad	6, 2	6, 3
bc	3, 3	7, 2
bd	1, 1	9, 5

A *behaviour strategy* of a player assigns a probability distribution on the moves for each information set of this player. Obviously a pure strategy is a special behaviour strategy.

Kuhn (1953) has proved an important theorem for games with perfect recall: Any mixed strategy of a player can be replaced by a behaviour strategy which is *realization equivalent* to the mixed strategy, that is, all nodes of the game tree are reached with the same probability, given any fixed strategies of the other players.

Theorem 1.3 (Kuhn's theorem) *For a player with perfect recall, any mixed strategy is realization equivalent to a behaviour strategy.*

The sequence form representation for an extensive game has been introduced by von Stengel (1996). For player i and any node t of the game tree, the sequence of moves of player i on the path from the root to the node t is uniquely determined, and is denoted by $\sigma_i(t)$. The empty sequence is denoted by $\emptyset$. Since we focus on perfect recall extensive games in this dissertation, for any nodes s and t that are in the same information set of player i, we have $\sigma_i(s) = \sigma_i(t)$; this is the definition of perfect recall used by Selten (1975), which is equivalent to the definition by Kuhn (1953). We denote by ξ_i the set of all sequences of player i, and by $\xi = \prod_{i \in P} \xi_i$ the set of sequence profiles.

For the extensive game in Example 1.1, the sets of sequences are

$$\xi_1 = \{\emptyset, A, B, AC, AD, AE, AF\}$$

for player 1 and

$$\xi_2 = \{\emptyset, a, b, c\}$$

for player 2. The sequence form payoffs are given for those sequence pairs that lead to a leaf of the game tree, with its corresponding payoff pair. Here they are given as

1 \ 2	$\emptyset$	a	b	c
$\emptyset$				
A		8 0		
B		7 7	2 3	4 5
AC			7 9	
AD			3 5	
AE				1 1
AF				4 9

where empty boxes correspond to pairs of sequences that do not arise from plays and have payoff entries zero for each player.

For the extensive game in Example 1.2, the sets of sequences are

$$\xi_1 = \{\emptyset, L, R\}$$

for player 1 and

$$\xi_2 = \{\emptyset, a, b, c, d\}$$

for player 2. The sequence form payoffs represent contributions to expected payoffs, using the chance probabilities, and are given by

1 \ 2	$\emptyset$	a	b	c	d
$\emptyset$					
L		4, 1	−1, 0	4, 3	2, 1
R		2, 0	5, 2	2, 0	4, 3

Compared with the strategic form and reduced strategic form that are both possibly exponential in the size of the game tree, the number of *sequences* of a game is bounded by the number of nodes of the game tree. Therefore, the sequence form is a polynomial-sized representation of the extensive game. However, randomizing between sequences can no longer be described by a single probability distribution, but requires a system of linear equations. We cover this in the next section.

1.3 The sequence form

In this section, we explain the extensive game sequence form representation of systems of linear equations. We explore how the theory of linear programming can be applied to the enumeration of equilibria for extensive games in the next section.

We recall some notation and propositions of the sequence form for two-player extensive games with perfect recall. For $i = 1,2$, the set of information sets of player i is denoted by H_i. For any information set $h \in H_i$, the set of moves at h is denoted by C_h. Let ξ_i be the set of sequences of player i. Since player i has perfect recall, any sequence σ in ξ_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,

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

where σ_h denotes the unique sequence leading to the information set h in H_i, defined by $\sigma_h = \sigma_i(t)$ for any t in h.

The number of sequences is given by

$$\begin{aligned} |\xi_i| &= |\{\emptyset\}| + |\{\sigma_h c \mid h \in H_i, c \in C_h\}| \\ &= 1 + \textstyle\sum_{h \in H_i} |C_h| \end{aligned}$$

which is polynomial in the size of the game tree. We denote by m and n the number of sequences for player 1 and 2 respectively, and let $k = |H_1| + 1$ and $l = |H_2| + 1$.

A *realization plan* for player i assigns a *realization probability* to every sequence in ξ_i. A realization probability is not a probability distribution, but satisfies a set of linear equalities that are stated in the following theorem:

Theorem 1.4 (von Stengel 1996) *For player 1, x is the realization plan of a mixed strategy if and only if $x(\sigma) \geq 0$ for all $\sigma \in \xi_1$ and*

$$x(\emptyset) = 1, \quad \sum_{c \in C_h} x(\sigma_h c) = x(\sigma_h) \quad \textit{for all } h \in H_1. \tag{1.1}$$

Similarly, y is the realization plan of a mixed strategy of player 2 if it is nonnegative and

$$y(\emptyset) = 1, \quad \sum_{c \in C_h} y(\sigma_h c) = y(\sigma_h) \quad \textit{for all } h \in H_2. \tag{1.2}$$

Equivalently, we can use matrix inequalities to express conditions (1.1) and (1.2):

$$Ex = e, \; x \geq \mathbf{0} \quad \text{and} \quad Fy = f, \; y \geq \mathbf{0}. \tag{1.3}$$

A sequence σ of player i *leads* to a leaf t of the game tree if $\sigma = \sigma_i(t)$. Sequence form *payoffs* are defined for pairs of sequences that lead to a leaf of the game tree, given by the payoff at the leaf times the probabilities of chance moves (if any) on the path to the leaf.

The sequence form payoffs are two sparse matrices A and B of dimension $|\xi_1| \times |\xi_2|$ for player 1 and 2, respectively. The expected payoffs under the realization plans x and y are $x^\top Ay$ and $x^\top By$, for player 1 and player 2, respectively. We consider the sequence form for the game trees in Examples 1.1 and 1.2.

Example 1.5 (Constraint matrices and payoff matrices) Consider the game tree in Example 1.1. The pair of sequences (A,a) leads to a leaf. None of the pairs (AC,a), (AD,a), (AE,a) or (AF,a) leads to a leaf.

The constraints (1.3) for this extensive game are

$$\begin{array}{c} \begin{array}{ccccccc} \emptyset & A & B & AC & AD & AE & AF \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} x = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \end{array}$$

$$\begin{array}{c} \begin{array}{cccc} \emptyset & a & b & c \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 1 & 1 \end{pmatrix} y = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ x \geq \mathbf{0},\ y \geq \mathbf{0}. \end{array}$$

The sparse payoff matrices are

$$A = \begin{pmatrix} & & & & & & \\ & 0 & 7 & & & & \\ & & 3 & 9 & 5 & & \\ & & 5 & & & 1 & 9 \end{pmatrix}$$

$$B = \begin{pmatrix} & & & & & & \\ & 8 & 7 & & & & \\ & & 2 & 7 & 3 & & \\ & & 4 & & & 1 & 4 \end{pmatrix}.$$

For the extensive game in Example 1.2, the constraint matrices are

$$E = \begin{pmatrix} 1 & & \\ -1 & 1 & 1 \end{pmatrix}, \quad e = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

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

and the sparse payoff matrices are

$$A = \begin{pmatrix} & & & & \\ & 4 & -1 & 4 & 2 \\ & 2 & 5 & 2 & 4 \end{pmatrix}, \qquad B = \begin{pmatrix} & & & & \\ & 1 & 0 & 3 & 1 \\ & 0 & 2 & 0 & 3 \end{pmatrix}.$$

1.4 The LCP for the sequence form

In this section, we discuss the theory of linear complementary problems (LCPs) applied to the enumeration of equilibria of two-player extensive games with perfect recall. The LCP that describes the set of Nash equilibria of such a game using the sequence form has been introduced by Koller, Megiddo, and von Stengel (1996). In our exposition, we use the notation of von Stengel (2002).

Theorem 1.6 *Consider the sequence form of a two-player extensive game with perfect recall. Let A, B be the payoff matrices and E, F be the constraint matrices for player 1 and player 2, respectively. The pair (x,y) of realization plans is a Nash equilibrium if and only if there are vectors u and v such that*

$$\begin{array}{ccccl} & & Ex & & = e \\ & & & Fy & = f \\ E^\top u & & & -Ay & \geq \mathbf{0} \\ & F^\top v & -B^\top x & & \geq \mathbf{0} \\ & & x, & y & \geq \mathbf{0} \end{array} \tag{1.4}$$

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

To get this result, consider the best response of player 1 against a given realization plan y of player 2. A best response realization plan of player 1 is the optimal solution of the following linear program (LP):

$$\begin{array}{ll} \text{maximize}_x & x^\top(Ay) \\ \text{subject to} & Ex = e \\ & x \geq \mathbf{0} \end{array} \tag{1.5}$$

Similarly, the best response realization plan of player 2 against a given realization plan x of player 1 is the optimal solution to

$$\begin{array}{ll} \text{maximize}_y & (x^\top B)y \\ \text{subject to} & Fy = f \\ & y \geq \mathbf{0} \end{array} \tag{1.6}$$

Thus a Nash equilibrium is a pair of realization plans (x,y) that is the optimal solution for the system (1.5) and (1.6). Unfortunately, by considering both x and y as variables, the objective functions are no longer linear. To avoid the non-linear system, the linear programming duality theorem is used to construct a new linear system.

We recall the LP duality theorem: Given the *primal* LP

$$\begin{array}{lll} \text{maximize} & \sum_{j=1}^{n} c_j x_j & \\ \text{subject to} & \sum_{j=1}^{n} a_{ij} x_j \leq b_i & (i = 1,2,\ldots,m), \\ & x_j \geq 0 & (j = 1,2,\ldots,n), \end{array} \tag{1.7}$$

its *dual* LP is

$$\begin{array}{lll} \text{minimize} & \sum_{i=1}^{m} b_i y_i & \\ \text{subject to} & \sum_{i=1}^{m} y_i a_{ij} \geq c_j & (j = 1,2,\ldots,n), \\ & y_i \geq 0 & (i = 1,2,\ldots,m). \end{array} \tag{1.8}$$

The duality theorem can be stated as follows:

Theorem 1.7 (LP duality theorem) *If the LP (1.7) has an optimal solution x, then the dual LP (1.8) also has an optimal solution y, and*

$$c^\top x = b^\top y.$$

Here $c = (c_j)_{j=1,\ldots,n}$ and $b = (b_i)_{i=1,\ldots,m}$.

In addition, the following *complementary slackness* conditions characterize a pair x,y of primal and dual solutions as optimal.

Theorem 1.8 (Complementary slackness) *A feasible solution x of the LP (1.7) is optimal if and only if there is an m-vector y such that*

$$\sum_{i=1}^{m} a_{ij} y_i = c_j \quad \textit{when} \quad x_j > 0$$
$$y_i = 0 \quad \textit{when} \quad \sum_{j=1}^{n} a_{ij} x_j < b_i$$

and

$$\sum_{i=1}^{m} a_{ij} y_i \geq c_j \quad \textit{for all} \quad j = 1, 2, \ldots, n,$$
$$y_i \geq 0 \quad \textit{for all} \quad i = 1, 2, \ldots, m.$$

Proofs of Theorems 1.7 and 1.8 can be found in Chvátal (1983, pp. 54–65).

Now apply Theorem 1.8 to the LPs (1.5) and (1.6) that describe the set of all Nash equilibria of an extensive game. The respective dual systems are

$$\begin{array}{ll} \text{minimize}_u & e^\top u \\ \text{subject to} & E^\top u \geq Ay \end{array} \tag{1.9}$$

and

$$\begin{array}{ll} \text{minimize}_v & f^\top v \\ \text{subject to} & F^\top v \geq B^\top x \end{array} \tag{1.10}$$

and the complementary slackness conditions are

$$x^\top (E^\top u - Ay) = 0 \tag{1.11}$$

and

$$y^\top (F^\top v - B^\top x) = 0. \tag{1.12}$$

This proves Theorem 1.6.

1.5 Extensive form correlated equilibrium

The concept of *correlated equilibrium* (CE) for the strategic form has been introduced by Aumann (1974), to explore the properties of solutions for games with communication. A correlated equilibrium is a Nash equilibrium of a game in which players can condition their action on payoff-irrelevant messages received before the game.

For a correlated equilibrium in *canonical form*, the messages can be seen as private recommendations for each player of what strategies to play. These recommendations can be correlated across the players. A correlated equilibrium can be defined as a probability distribution over the set of strategy profiles such that, when a *correlation device* draws a strategy profile according to this probability distribution and sends the strategies to the corresponding players privately as recommendations, the players do not have an incentive to deviate to any other strategy – assuming that the other players conform to the recommendation they get.

Formally, consider a finite strategic form game $\Gamma = (P, (S_p)_{p \in P}, (a^p)_{p \in P})$. The set of players is P. The set of pure strategies of player p is S_p. Let $S = \prod_{p \in P} S_p$ be the set of strategy profiles. The payoff function of player p is $a^p : S \to \mathbb{R}$. A *correlated equilibrium* is a distribution x on S such that for all players p and all pairs of strategies $i, j \in S_p$

$$\sum_{s_{-p} \in S_{-p}} \big[a^p(i, s_{-p}) - a^p(j, s_{-p})\big] x(i, s_{-p}) \geq 0 \tag{1.13}$$

where (i, s_{-p}) is the strategy profile resulting from $s_{-p} \in S_{-p}$ by adding the component $i \in S_p$. The inequalities (1.13) are the *incentive constraints* that characterize the equilibrium property. For every pair of strategies $i, j \in S_p$, the sum of products $\sum_{s_{-p} \in S_{-p}} [a^p(i, s_{-p}) - a^p(j, s_{-p})] x(i, s_{-p})$ is the expected payoff player p gets by playing i. It is as large as that by playing j, conditional on the recommended strategy being i.

It is easy to verify that a Nash equilibrium is a correlated equilibrium that happens to be a product distribution, that is, a distribution x on S so that there is a distribution x^p on S_p for each $p \in P$ and so that $x(s) = \prod_{p=1}^{n} x^p(s_p)$ for all $s = (s_1, \ldots, s_n) \in S$.

The original approach by Aumann (1974) to correlated equilibria is more detailed. The *correlated equilibrium* refers to the Nash equilibrium of the extended game γ_d where players are first informed by the correlated device of their private messages and then play γ. The probability distribution on strategy profiles that is induced by a correlated equilibrium is called a *correlated equilibrium distribution*.

The difference between the definition of correlated equilibrium and correlated equilibrium distribution is important when the mere revelation of strategies fails for some concepts. For instance, Dhillon and Mertens (1996) show that this direct mechanism does not yield a "perfect" correlated equilibrium. However, for us signals can be recommended actions and so we use the shorter word correlated equilibrium for correlated equilibrium distribution without ambiguity.

Consider the (strategic-form) correlated equilibrium for an extensive game with perfect recall. A pure strategy of a player in an extensive game specifies one move at each of this player's information sets. The *posterior* of a player p at any stage of the game is the probability of the other players' recommendations, based on the information player p gets. There are two ways to extend the definition of correlated equilibrium to extensive form games. One is to interpret the extensive form as being merely a concise description of a strategic form game and apply the concepts described above to this strategic form game. In this case, the recommendations are generated before the beginning of the game and are sent to the corresponding players immediately.

Alternatively, one may consider a definition of correlated equilibrium where the players receive *delayed* recommendations. That is, the strategy profile selected according to the device defines a move c for each information set h of each player p, which is sent to player p only when he reaches h. It is optimal for the player to follow the recommended move, assuming that all other players follow their recommendations. The advantage, compared to the CE for the strategy form, is that the player only has to compare possible alternative moves at a time, rather than alternative strategies. As long as the player follows the recommendation, he gets further information as he reaches a subsequent information set, thus his posterior changes as the game proceeds. This extension of correlated equilibrium has been proposed and explored by von Stengel and Forges (2008), as extensive form correlated equilibrium. They also show (p. 1005) that the set of EFCE does not change when the player does not get any further information whenever he deviates at some information set, which means that the strategies chosen as recommendations can be reduced strategies; we always make this assumption. The following is the formal definition of an EFCE.

Definition 1.9 *Given a correlation device μ, consider the extended game in which a chance move first selects a strategy profile π according to μ. Then, whenever a player p reaches an information set h in H_p, he receives the move c at h specified in π as a signal, interpreted as a recommendation to play c. An* extensive form correlated equilibrium (EFCE) *is a Nash equilibrium of such an extended game in which the players follow their recommendations.*

As recommendations become local to each information set, the players know less than in the CE where players get all recommendations before the game starts, so the equilibrium conditions are easier to satisfy. As a result the set of EFCE is larger than the set of CE. Von Stengel (2001) gives an example of an extensive form game which has an EFCE that is not in the set of CE. Another example showing this is given by Myerson (1986).

Example 1.10 (Myerson 1986) Consider the game tree in Figure 1.3.

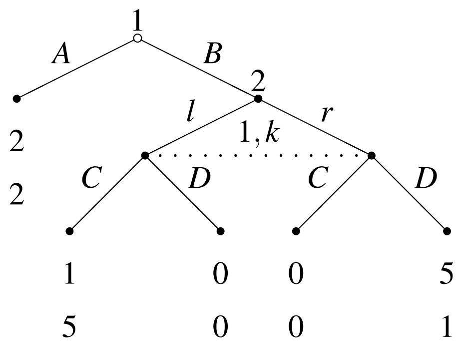


Figure 1.3: Extensive game with an EFCE that is not a CE.

The correlated device that selects (BC, l) and (BD, r) with probability $\frac{1}{2}$, and all other strategy profiles with probability 0 is an EFCE (with expected payoff 3 for each player) but not a strategic form correlated equilibrium. If player 1 gets the recommended strategy, say, BC before the game, he knows that his opponent gets l

as recommendation for sure. Thus player 1 prefers to deviate to $A*$ to get a higher payoff of 2, instead of 1 from BC.

The set of correlated equilibrium distributions is a convex polyhedron, unlike the set of Nash equilibria. We now list a few results of the complexity of correlated equilibrium computation.

For the strategic form, based on a variant of the existence proof due to Hart and Schmeidler (1989), Papadimitriou and Roughgarden (2008) develop an algorithm for finding correlated equilibria in a class of succinctly representable multi-player games, including graphical games, anonymous games, polymatrix games, congestion games, scheduling games, local effect games and several generalizations. They employ linear programming duality, the ellipsoid algorithm, Markov chain steady state computations, and application-specific methods for computing multivariate expectations over product distributions, to compute a correlated equilibrium for such games in polynomial time.

Theorem 1.11 (Papadimitriou and Roughgarden 2008) *Consider a game G in some description so that the number of players and the number of strategies per player is polynomial in the size of the description, and so that the expected payoff for any product distribution (mixed strategy profile) can be computed in polynomial time. Then one can compute in polynomial time a CE of G that is a convex combination of polynomially many product distributions.*

Their algorithm is based on an assertion that if the ellipsoid algorithm is run on an infeasible linear program, and the resulting sequence of cuts are combined to form a new linear program, then this new LP will also be infeasible because the ellipsoid algorithm would run the same way on it. This is called the “Ellipsoid Against Hope” algorithm. As Stein et al. (2010) point out, the problem with the reasoning of this algorithm is that the coefficients of the new LP may be larger, requiring a bigger initial bounding ellipsoid when running the ellipsoid algorithm on it. They construct a counterexample which is a game with a unique irrational exchangeable equilibrium. Since the steps of Papadimitriou and Roughgarden’s algorithm imply that a rational exchangeable equilibrium can be computed, the algorithm fails when applied to this game. Jiang and Leyton-Brown (2010) present a variant of the Ellipsoid Against Hope algorithm that they called the "Simplified Ellipsoid Against Hope" algorithm. They prove that this algorithm overcomes the precision issue and computes a CE in polynomial time.

For the strategic form correlated equilibrium of an extensive game, von Stengel (2001) and Chu and Halpern (2001) proves that the set of CE cannot be computed in polynomial time, unless P=NP. Here polynomial (or linear or exponential) size and time always refers to the size of some description of the extensive game with its game tree, information sets, moves, chance probabilities, and payoffs.

Theorem 1.12 (von Stengel 2001, Chu and Halpern 2001) *For extensive two-player games with perfect recall and without chance moves, it is NP-hard to find a strategic form correlated equilibrium with maximum payoff sum.*

In contrast, von Stengel and Forges (2008) prove that the set of EFCE for a two-player perfect-recall extensive game without chance moves has a polynomial-sized description.

Theorem 1.13 (von Stengel and Forges 2008) *For a two-player, perfect-recall extensive game without chance moves, the set of EFCE can be described by a system of linear equations and inequalities of polynomial size. For any solution to that system (which defines an EFCE), a pair of pure strategies containing the recommended moves can be sampled in polynomial time.*

Von Stengel and Forges (2008) also show that for games with chance moves, the problem of computing a maximum-sum EFCE is NP-hard.

However, it can be easier to compute one EFCE. For multi-player extensive games with perfect recall, we show that the problem of finding one EFCE is polynomial.

Theorem 1.14 (Computing one EFCE in polynomial time) *For a multi-player extensive game with perfect recall, one EFCE can be found in polynomial time.*

This is the main result of Chapter 3 of this thesis.

CHAPTER 2

Equilibrium Enumeration Using the Sequence Form

This chapter focuses on equilibrium enumeration for two-player extensive games. Using the sequence form that we reviewed in Section 1.3, we start in Section 2.1 with simplifying the system (1.4) in Section 1.4 that describes the set of Nash equilibria. The resulting system given in Section 2.2 is less elegant but has fewer variables and constraints. We review in Section 2.3 algorithms for computing equilibria of bimatrix games. In Section 2.4, we discuss the polyhedra that result from the sequence form. In Section 2.5, we show how a classic equilibrium enumeration algorithm based on vertex enumeration can be extended to apply to extensive games. In Section 2.6, we extend the (different) EEE algorithm to extensive games, and conclude with some examples.

The following is a summary of the novel contributions of this chapter.

First, we present *a reduced system* for the set of Nash equilibria of extensive games. We categorize the sequences into three groups: terminal independent sequences, terminal dependent sequences and non-terminal sequences, and study the relations between the three kinds of sequences. Then we remove the redundant constraints for the *realization probabilities* of the non-terminal sequences. Compared with the system by von Stengel (1996) that also uses the sequence form representation, our system after removing the redundancy has fewer variables and constraints, and is a reduced system to describe the set of Nash equilibria.

Second, we present (and implemented) an algorithm for finding all Nash equilibria of extensive games based on vertex enumeration. The algorithm we use to find all Nash equilibria of extensive games is analogous to that for the strategic form. We prove that for any Nash equilibrium of the extensive game, there is a pair of vertices representing this Nash equilibrium. The *lrs* algorithm by Avis and Fukuda (1992) and integer pivoting, are used for the implementation of this algorithm.

Third, we extend the EEE equilibrium enumeration algorithm to two-player extensive games. The EEE algorithm for bimatrix games, for "enumeration of extreme equilibria", is due to Audet, Hansen, Jaumard, and Savard (2001). Several improvements to this algorithm, as well as putting it in the context of vertex enumeration, have been presented by Avis Rosenberg, Savani, and von Stengel (2010). Both the EEE and improved EEE algorithm, as well as the algorithm based on vertex enumeration mentioned before, use a pair of best response polyhedra. We introduce a pair of polyhedra for the reduced system for the extensive game and label the points in these polyhedra to identify the complementary slackness condition. These polyhedra are analogous to the best response polyhedra to identify the possible supports of strategies for bimatrix games. We prove that the EEE algorithm applied to these polyhedra computes all extreme Nash equilibria for a two-player extensive game.

2.1 Removing redundant variables and constraints

In Chapter 1 we explained, using the duality theorem, how the set of Nash equilibria can be described as the solutions of system (1.4) that satisfy the complementary slackness conditions. The LCP (1.4) has nonnegative variables x, y that are subject to the equations $Ex = e$ and $Fy = f$. The goal of this section is to use these equations to remove the *redundant variables* and *redundant constraints* in the system (1.4).

To explain the redundant variables and redundant constraints, we first categorize the sequences into two groups: terminal sequences and non-terminal sequences. As we explained in Section 1.3, the realization plans satisfy certain constraints. Redundancy arises because of the constraint equalities. The realization probabilities assigned to sequences that are determined by the realization probabilities of certain other sequences are redundant variables.

We define a *terminal sequence* as follows.

Definition 2.1 *A sequence* σ *is called a* terminal sequence *if and only if there is no move extending* σ *to a longer sequence. Otherwise it is called a* non-terminal sequence.

As an example, the terminal sequences of the game in Example 1.1 are B, AC, AD, AE and AF for player 1 and a, b and c for player 2. All these sequences lead to leaves.

A terminal sequence σ leads to a leaf, but not all sequences that lead to a leaf are terminal. Some non-terminal sequences can still lead to leaves.

Example 2.2 (A non-terminal sequence leading to a terminal node) Consider the game tree in Figure 2.1 where the payoffs are omitted. Here the sequence R of player 1 leads to a terminal node with the sequence r of player 2. The terminal node is reached by a move of player 2. When player 1 chooses R and player 2 chooses l, the subsequent decision node of player 1 is parallel to the terminal node that R leads to. All the moves that player 1 can choose at that decision node can extend the sequence R. Thus R is a non-terminal sequence that leads to a terminal node.

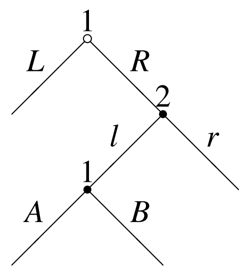


Figure 2.1: An extensive game with a non-terminal sequence, here R for player 1, which leads to a terminal node.

In the constraint matrices E and F, columns without -1 as an entry correspond to terminal sequences; other columns correspond to non-terminal ones.

In this chapter, we denote m and n to the numbers of sequences (including the empty sequence) for player 1 and player 2, respectively. We show that the k equations $Ex = e$ are k linearly independent equations for m variables, where k is defined in Section 1.3 as the number of information sets plus one. Therefore it is possible to express k of these variables, called dependent variables, as linear functions of the other $m-k$ variables, which are called independent. We call the k variables "redundant variables" because it is not necessary to compute these variables when all the other $m-k$ independent variables are known. Substituting linear expressions into the other constraints reduces the number of variables for player 1 by k. Furthermore, we show that it is possible to express the realization probabilities of all the non-terminal sequences as linear combinations of realization probabilities of certain terminal sequences. So the variables for non-terminal sequences are all redundant variables.

The nonnegativity constraints for the dependent variables are normally still necessary. When deriving the corresponding LCP via linear programming as in Section 1.4, this leads to k dual variables. As we will show, some nonnegativity constraints for the components of x are implied by the others because of the equations (1.1). Namely, the realization probability of any non-terminal sequence is the sum of the realization probabilities of sequences extended by moves at the information set that the non-terminal sequence leads to. Moreover, we show that the realization probability of any non-terminal sequence is the sum of realization probabilities of certain terminal sequences. Thus the nonnegativity constraints for the non-terminal sequences are satisfied if the realization probabilities of all the terminal sequences are nonnegative. We say these nonnegativity constraints for the non-terminal sequences are redundant constraints because they can be eliminated from the system without changing the solution set of the system. The nonnegativity constraints for terminal sequences are not redundant and must be stated explicitly.

The following lemma shows that the constraints for non-terminal sequences are redundant.

Lemma 2.3 *For every non-terminal sequence σ_h there are some terminal sequences σ_i so that*

$$x(\sigma_h) = \sum_i x(\sigma_i).$$

Proof. Suppose the above is not true. Since the game is finite, we can find a non-terminal sequence σ_h with the largest number of moves, for which the realization probability cannot be written as the sum of the realization probabilities of some terminal sequences.

Since σ_h is non-terminal, it can be extended to longer sequences $\sigma_h c$ and

$$x(\sigma_h) = \sum_{c \in C_h} x(\sigma_h c).$$

There must be at least one non-terminal sequence σ_{hc^*} whose realization probability cannot be written as $\sum_i x(\sigma_i^{c^*})$ where $\sigma_i^{c^*}$ are some terminal sequences. Otherwise $x(\sigma_h) = \sum_{c \in C_h} x(\sigma_h c) = \sum_{c \in C_h} \sum_i x(\sigma_i^c)$ which is the sum of the realization probabilities of some terminal sequences. That is a contradiction.

Thus σ_{hc^*} is a longer sequence for which the realization probability cannot be written as the sum of the realization probabilities of some terminal sequences. But we have assumed σ_h is the longest one. So it is a contradiction. Therefore the above statement is true. □

To identify the redundant variables, we categorize the sequences into *dependent sequences* and *independent sequences*.

Given an extensive game, consider the two system of linear equations, $Ex = e$ and $Fy = f$, for the two players. These systems are both consistent, and in most cases both have infinitely many solutions. This holds because both systems have fewer equations than unknowns. The only exception arises when a player has only information sets with a trivial single move (or no information sets at all). Otherwise, observe that for each system, the number of equations for player i (that is, k for player 1 and l for player 2) is equal to $|H_i| + 1$, that is, the number of information sets of that player plus 1. The number of unknowns for player i is equal to the number of sequences (that is, m for player 1 and n for player 2), which is $\sum_{h \in H_i} |C_h| + 1$. Every information set h has at least one move, that is, $|C_h| \geq 1$, so as soon as there is an information set h with more than one move, $|C_h| > 1$, we have more unknowns than equations. In this dissertation, we focus on non-trivial games with $k < m$ and $l < n$.

When the systems (1.3) are solved, to describe the infinitely many solutions, the variables are designated as either independent (or free) or dependent. The independent variables are those that are allowed to take any value (subject to inequality constraints), while the remaining (dependent) variables are dependent on the values of the independent variables. For a given description of solutions of the system (1.3), we say the sequences whose realization probabilities are independent are *independent sequences*, and the other sequences are called *dependent sequences*.

For a system of linear equations $Ax = b$, the number of dependent variables is equal to the rank of the matrix A. We now show that the ranks of the constraint matrices E and F are full, that is, k and l, which are 1 more than the number of information sets of the respective player. We also show how to identify the k and l independent columns for E and F, respectively.

Consider the constraint matrix E. One of the rows is $x(\emptyset) = 1$. Each of the remaining rows $\sum_{c\in C_h} x(\sigma_h c) = x(\sigma_h)$ corresponds to an information set h of player 1, and is written as

$$-x(\sigma_h) + \sum_{c\in C_h} x(\sigma_h c) = 0. \tag{2.1}$$

Of course, (2.1) could also be written with opposite signs for all coefficients, but this way every sequence $\emptyset$ or $\sigma_h c$ for some $h \in H_1$ and $c \in C_h$ appears exactly once with coefficient 1 in a column of E. We order the rows of E so that:

1. the first row of E is $x(\emptyset) = 1$, and
2. for any $i, j \leq k$, if h_i is the information set of the ith row and h_j is the information set of the jth row of E, and h_i precedes h_j, then $i < j$.

Lemma 2.4 *For each row i of the matrix E with rows in the described order, written as (2.1), there is a column so that in that column: (i) the ith row entry is* 1*, and (ii) for any* $0 \leq j < i$*, the jth row entry is 0.*

Proof. First we show that the first non-zero row entry of each column is 1, and all other non-zero row entries are -1. Each column of E corresponds to a sequence of player 1. If the sequence of the column is $\emptyset$, then the first row entry is also the first non-zero row entry, and it is 1. If the sequence of the column is $\sigma_h c$ for some h and move c at h, then the row entry for h is 1. All other non-zero row entries correspond to information sets that the sequence leads to. So h precedes these information sets, and the indices of these non-zero rows are all greater than that of the row of h. That is, the first non-zero row entry is in the row of h, and the value of it is 1. The value of any other non-zero row entry is -1.

For each row corresponding to information set h, there is at least one move c at h. Consider the column of the sequence $\sigma_h c$. The row entry for h of this column is 1. It is the first non-zero row entry of this column. □

Fix any choice of k columns according to Lemma 2.4. These define a lower triangular matrix with 1's on the diagonal and zeros above the diagonal. So the rank of E is k. Similarly, the rank of F is l.

For clarity, we note that the concepts of independent variables and independent columns are different. The matrix E has rank k, which means that there are k linearly independent columns, and all remaining columns are "dependent" in the sense that they can be expressed as linear combinations of these k columns. The independent columns do not correspond to the independent variables. On the contrary, for the system $Ex = e$, there is a description for any solution such that all the variables corresponding to the k independent columns are designated dependent variables and all the other $n-k$ variables are independent variables. This applies similarly to the system $Fy = f$.

Therefore, we can use the k independent columns to find $n-k$ independent sequences (or independent variables). The $n-k$ sequences (or variables) that corresponds to the remaining columns are the independent sequences (or independent variables). Let E_B be a square matrix made up from the k linearly independent columns of E, and let $E = [E_B \; E_I]$. So E_B and E_I are the columns corresponding to the dependent variables and independent variables, respectively. The constraints $Ex = e$ become

$$[E_B \; E_I] \begin{bmatrix} x_B \\ x_I \end{bmatrix} = e,$$

where the components of x_B correspond to the columns in E_B, and thus are dependent variables, and the components of x_I are the independent variables. Thus $Ex = e$

is written as

$$E_B x_B + E_I x_I = e \tag{2.2}$$

where E_B is invertible, and the dependent variables x_B are given by

$$x_B = E_B^{-1}(e - E_I x_I).$$

The set of k independent columns is not uniquely determined. There may be more than one set such that all k columns are independent. For convenience, we choose the sets B and I so that all variables for the non-terminal sequences are in x_B. We can always do this according to the following lemma.

Lemma 2.5 *In the constraint matrices E and F, the columns of all non-terminal sequences are linearly independent.*

Proof. The last non-zero row entry of any column of a non-terminal sequence is -1. This holds because each nonempty sequence is uniquely determined by its last move at a unique information set, so there is only one 1 in each column. By Lemma 2.4, this row of 1 is the first non-zero row. By definition, there is at least one information set that the non-terminal sequence leads to, and thus at least one -1 in a column of the non-terminal sequence.

For each row of an information set, there is only one -1, since this information set is reached by only one sequence. Therefore, the indices of the last non-zero rows of columns of non-terminal sequences are all different. So none of these columns can be expressed as a linear combination of the other columns. That is, all of these columns are linearly independent. □

Let k_n denote the number of non-terminal sequences. Then we can find the remaining $k - k_n$ columns by iteration. For initialization, let E_B be the columns of all the non-terminal sequences. In each iteration, choose one column that does not belong to E_B. Check whether the column and the columns in E_B are independent. If so, add this column to E_B. Otherwise go to the next iteration, until the number of columns in E_B reaches k, which is the rank of E. So E_B is the required matrix. Alternatively, we can find E_B by Gaussian elimination, in time $O(n^3)$. When using

Gaussian elimination to find E_B, we need to order the columns so that for any index $0 < i \leq k$ and $0 < j \leq l$, if the sequence of the ith column is non-terminal and that of the jth column is terminal, then $i < j$. That is, we make sure that columns for non-terminal sequences are on the left of the matrix and columns for terminal sequences on the right. Doing this ordering guarantees that columns of all non-terminal sequences are in E_B.

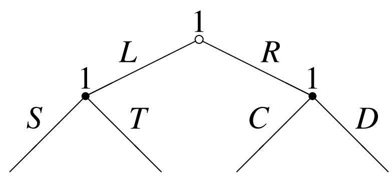


Figure 2.2: Extensive game, without payoffs, to show choices of independent sequences.

Figure 2.2 gives an example that demonstrates that the described way of choosing all non-terminal sequences as part of the independent columns in E_B is *not* the algorithmically most straightforward way of finding k linearly independent columns of E. For the game in Figure 2.2, the constraints $Ex = e$ are

$$\begin{array}{ccccccc} \emptyset & L & R & LS & LT & RC & RD \end{array}$$
$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 1 \end{pmatrix} x = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

The most straightforward way of finding k independent columns of E would be to select in each row the first column with a 1 in it. In this example, these would give the colums $\emptyset, L, LS, RC$. This selection gives a lower triangular submatrix E_B of E that is easy to invert. The resulting row operations define the following system, with the columns of the matrix E_B (which becomes the identity matrix) underlined:

$$\begin{array}{ccccccc} \underline{\emptyset} & \underline{L} & R & \underline{LS} & LT & \underline{RC} & RD \end{array}$$
$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 1 \end{pmatrix} x = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}.$$

However, the non-terminal sequence R does not define a column of E_B, contrary to our intention.

In Figure 2.2, using the columns $\emptyset, L, R, LS$ to define E_B includes all non-terminal sequences as intended. The resulting system after row operations that convert E_B to an identity matrix is

$$\begin{array}{ccccccc} \underline{\emptyset} & \underline{L} & \underline{R} & \underline{LS} & LT & RC & RD \end{array}$$
$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} x = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}.$$

We explain the construction of E_B in more detail for the game tree in Figure 1.1.

Example 2.6 (Finding independent variables) Consider the game tree in Figure 1.1. The constraint matrices E and F are shown in Example 1.5. Let the independent columns of E be defined by the sequences $\emptyset, A, B, AC$ and those of F by $\emptyset, a$. After the row operations, the two systems of constraints become

$$\begin{array}{ccccccc} \underline{\emptyset} & \underline{A} & \underline{B} & \underline{AC} & AD & AE & AF \end{array}$$
$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & -1 & -1 \end{pmatrix} x = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$\begin{array}{cccc} \underline{\emptyset} & \underline{a} & b & c \end{array}$$
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

The solutions can be described as

$$\begin{array}{llllllll} x_\emptyset & & & & = & 1 & & \\ & x_A & & & = & & x_{AE} & +x_{AF} \\ & & x_B & & = & 1 & -x_{AE} & -x_{AF} \\ & & & x_{AC} & = & -x_{AD} & +x_{AE} & +x_{AF} \end{array}$$

and

$$\begin{array}{llllll} y_\emptyset & & = & 1 & & \\ & y_a & = & 1 & -y_b & -y_c\,. \end{array}$$

The independent sequences are AB, AE, AF for player 1 and b, c for player 2. As required, all the independent sequences are terminal sequences.

This is not the only choice of set of independent sequences. The following shows another choice of E_B for player 1 with a different set of independent sequences:

$$\begin{array}{ccccccc} \underline{\emptyset} & A & \underline{B} & \underline{AC} & AD & \underline{AE} & AF \end{array}$$
$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} x = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

The solution is described as

$$\begin{array}{lllllll} x_\emptyset & & & & = & 1 & & & \\ & x_B & & & = & 1 & -x_A & & \\ & & x_{AD} & & = & & x_A & -x_{AC} & \\ & & & x_{AE} & = & & x_A & & +x_{AF} \end{array}$$

The independent sequences are A, AC, AF for player 1. Here, A is an independent non-terminal sequence. As mentioned, we do not use this description for our algorithm to find Nash equilibria.

2.2 The system after removing redundancy

As we explained in the last section, all independent variables in x_I correspond to terminal sequences. Dependent variables in x_B correspond to either terminal sequences or non-terminal sequences. Let $x_B = [x_N\ x_D]$ where all non-terminal sequence variables are in x_N and all terminal sequence variable components of x_B are in x_D.

So there are three kinds of sequences:

1. Terminal sequences whose realization probabilities x_I are independent;

2. Terminal sequences whose realization probabilities x_D are dependent, which can be expressed as $p_2 + P_2 x_I$;

3. Non-terminal sequences whose realization probabilities x_N are dependent, which can be expressed as $p_1 + P_1 x_I$, and for which the nonnegativity constraints are redundant, according to Lemma 2.3.

Here

$$\begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = E_B^{-1} e, \quad \begin{pmatrix} P_1 \\ P_2 \end{pmatrix} = -E_B^{-1} E_I. \tag{2.3}$$

We now introduce the new system describing the set of Nash equilibria after eliminating the redundant variables x_N and their nonnegativity constraints $x_N \geq 0$. The realization plans for player 1 and 2 are

$$\begin{aligned} x &= \begin{pmatrix} x_N \\ x_D \\ x_I \end{pmatrix} = \begin{pmatrix} p_1 \\ p_2 \\ 0 \end{pmatrix} + \begin{pmatrix} P_1 \\ P_2 \\ I_{m-k} \end{pmatrix} x_I, \\ y &= \begin{pmatrix} y_N \\ y_D \\ y_I \end{pmatrix} = \begin{pmatrix} q_1 \\ q_2 \\ 0 \end{pmatrix} + \begin{pmatrix} Q_1 \\ Q_2 \\ I_{n-l} \end{pmatrix} y_I. \end{aligned} \tag{2.4}$$

We substitute x and y as functions of x_I and y_I, respectively, according to (2.4), and remove the redundant nonnegativity constraints $x_N \geq \mathbf{0}$ and $y_N \geq \mathbf{0}$ for non-terminal sequences. Then the pairs (1.5) and (1.6) of LPs become:

$$\begin{aligned} \text{maximize}_{x_I} \quad & c_1 + x_I^\top (a' + A' y_I) \\ \text{subject to} \quad & -P_2 x_I \leq p_2 \\ & x_I \geq \mathbf{0} \end{aligned} \tag{2.5}$$

and

$$\begin{aligned} \text{maximize}_{y_I} \quad & c_2 + (b' + x_I^\top B')y_I \\ \text{subject to} \quad & -Q_2 y_I \le q_2 \\ & y_I \ge \mathbf{0} \end{aligned} \tag{2.6}$$

where

$$c_1 = \begin{pmatrix} p_1 \\ p_2 \\ 0 \end{pmatrix}^\top A(\begin{pmatrix} q_1 \\ q_2 \\ 0 \end{pmatrix} + \begin{pmatrix} Q_1 \\ Q_2 \\ I_{n-l} \end{pmatrix} y_I),$$

$$c_2 = (\begin{pmatrix} p_1 \\ p_2 \\ 0 \end{pmatrix} + \begin{pmatrix} P_1 \\ P_2 \\ I_{m-k} \end{pmatrix} x_I)^\top B \begin{pmatrix} q_1 \\ q_2 \\ 0 \end{pmatrix},$$

$$a' = \begin{pmatrix} P_1 \\ P_2 \\ I_{m-k} \end{pmatrix}^\top A \begin{pmatrix} q_1 \\ q_2 \\ 0 \end{pmatrix}, \qquad b' = \begin{pmatrix} p_1 \\ p_2 \\ 0 \end{pmatrix}^\top B \begin{pmatrix} Q_1 \\ Q_2 \\ I_{n-l} \end{pmatrix},$$

and

$$A' = \begin{pmatrix} P_1 \\ P_2 \\ I_{m-k} \end{pmatrix}^\top A \begin{pmatrix} Q_1 \\ Q_2 \\ I_{n-l} \end{pmatrix}, \qquad B' = \begin{pmatrix} P_1 \\ P_2 \\ I_{m-k} \end{pmatrix}^\top B \begin{pmatrix} Q_1 \\ Q_2 \\ I_{n-l} \end{pmatrix}.$$

The dual systems to (2.5) and (2.6) use vectors of dual variables u and v of the same dimensions as x_D and y_D, respectively, and are given by

$$\begin{aligned} \text{minimize}_u \quad & c_1 + p_2^\top u \\ \text{subject to} \quad & -P_2^\top u \ge a' + A' y_I \\ & u \ge \mathbf{0} \end{aligned} \tag{2.7}$$

and

$$\begin{aligned} \text{minimize}_v \quad & c_2 + q_2^\top v \\ \text{subject to} \quad & -Q_2^\top v \ge (b')^\top + (B')^\top x_I \\ & v \ge \mathbf{0} \end{aligned} \tag{2.8}$$

By strong duality, we have the following theorem for the reduced system that describes the set of Nash equilibria.

Theorem 2.7 *The pair (x,y) of realization plans as in (2.4) defines an equilibrium if and only if there are vectors u, v so that*

$$\begin{array}{rrrrl} & & P_2 x_I & & \geq -p_2 \\ & & & Q_2 y_I & \geq -q_2 \\ -P_2^\top u & & & -A' y_I & \geq a' \\ & -Q_2^\top v & -(B')^\top x_I & & \geq (b')^\top \\ u, & v, & x_I, & y_I & \geq \mathbf{0} \end{array} \tag{2.9}$$

and the following complementarity conditions hold:

$$\begin{aligned} u^\top(P_2 x_I + p_2) &= 0, \\ v^\top(Q_2 y_I + q_2) &= 0, \\ x_I^\top(-P_2^\top u - A' y_I - a') &= 0, \\ y_I^\top(-Q_2^\top v - B' x_I - (b')^\top) &= 0. \end{aligned} \tag{2.10}$$

Although the system (2.9) is less elegant than the original system (1.4), it is important for the equilibrium computation algorithms in the sense of reducing run time and memory space. We now illustrate this theorem by Example 1.1.

Example 2.8 (The reduced system describing the set of Nash equilibria) Consider the extensive game in Example 1.1. With the choice of independent sequences in Example 2.6,

$$x_I = (x_{AD}, x_{AE}, x_{AF}), \quad x_N = (x_\emptyset, x_A), \quad x_D = (x_B, x_{AC}),$$

$$y_I = (y_b, y_c), \quad y_N = (y_\emptyset), \quad y_D = (y_a)$$

and

$$x_N = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} x_I$$

$$x_D = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 & -1 & -1 \\ -1 & 1 & 1 \end{pmatrix} x_I$$

$$y_N = \begin{pmatrix} 1 \end{pmatrix} + \begin{pmatrix} 0 & 0 \end{pmatrix} y_I$$

$$y_D = \begin{pmatrix} 1 \end{pmatrix} + \begin{pmatrix} -1 & -1 \end{pmatrix} y_I$$

where

$$p_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad p_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

$$q_1 = q_2 = (1),$$

and

$$P_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \qquad P_2 = \begin{pmatrix} 0 & -1 & -1 \\ -1 & 1 & 1 \end{pmatrix},$$

$$Q_1 = (0\,0), \qquad Q_2 = (-1\ -1).$$

Only a', b', A' and B' are needed in Theorem 2.7. They are

$$a' = \begin{pmatrix} 0 \\ -7 \\ -7 \end{pmatrix}, \qquad b' = \begin{pmatrix} -5 - 3 \end{pmatrix},$$

and

$$A' = \begin{pmatrix} -4 & 0 \\ 13 & 3 \\ 13 & 11 \end{pmatrix}, \qquad B' = \begin{pmatrix} -4 & 0 \\ 4 & -4 \\ 4 & -1 \end{pmatrix}.$$

2.3 Review of equilibrium enumeration for bimatrix games

In this section we recall some algorithms for computing equilibria of bimatrix games. Although we focus on computing equilibria of two-player extensive games in this chapter, we explore the algorithms for bimatrix games for two reasons. First, by the strategic form representation, a two-player extensive game is of the form of a bimatrix game. Thus all algorithms for computing equilibria of bimatrix games can be applied to extensive games. Second, the algorithms computing equilibria of two-player extensive games we introduce in this chapter are extensions of the algorithms for bimatrix games in this section.

For a bimatrix game (A,B), we denote the sets of labels of strategies of player 1 and 2 by $\Phi = \{1,\ldots,\varphi\}$ and $\Psi = \{\varphi+1,\ldots,\varphi+\psi\}$, respectively, so that the two sets are disjoint. The payoff matrices A and B belong to $\mathbb{R}^{\Phi\times\Psi}$. For $i = 1,2$, a *best response* of player i against a certain strategy of the other player is a strategy of player i that maximizes player i's expected payoff.

The algorithms in this subsection are all based on the following best response condition by Nash (1951); our exposition follows Avis et al. (2010).

Theorem 2.9 (Best response condition, Nash 1951) *Let x and y be mixed strategies of player* 1 *and* 2*, respectively. Then x is a best response to y if and only if for all $i \in \Phi$,*

$$x_i > 0 \;\Rightarrow\; (Ay)_i = u = \max\{(Ay)_k \mid k \in \Phi\}. \tag{2.11}$$

and y is a best response to x if and only if for all $j \in \Psi$,

$$y_j > 0 \;\Rightarrow\; (B^\top x)_j = v = \max\{(B^\top x)_k \mid k \in \Psi\}. \tag{2.12}$$

We recall (following von Stengel 2002) some notions from the theory of convex polytopes (see Ziegler 1995). An *affine combination* of points $z_1,\ldots,z_k$ in some Euclidean space is of the form $\sum_{i=1}^k z_i\lambda_i$ where $\lambda_1,\ldots,\lambda_k$ are reals with $\sum_{i=1}^k \lambda_i = 1$. It is called a *convex combination* if $\lambda_i \geq 0$ for all i. A set of points is *convex* if it is closed under forming convex combinations. Given points are *affinely independent* if none of these points is an affine combination of the others. A convex set has *dimension* d if and only if it has $d+1$, but no more, affinely independent points.

A *polyhedron* P is of the form

$$\{y \in \mathbb{R}^d \mid Cy + b \geq \mathbf{0}\} \tag{2.13}$$

where C is some matrix and b a vector. It is called a *polytope* if it is bounded. A *face* of P is defined by making some of the inequalities in (2.13) *binding*, that is, considering them as equations. An *edge* is a 1-dimensional face. Suppose P has dimension d. A *facet* is a $(d-1)$-dimensional face of P, where exactly one inequality in (2.13) is binding. A *vertex* $y \in \mathbb{R}^d$ is a point of P that satisfies an independent set of d inequalities in (2.13) as equations, so a vertex is the unique element of a 0-dimensional face of P. An *extreme ray* $z \in \mathbb{R}^d$ is a direction such that for some vertex y and any positive number α, the point $y + \alpha z$ is in P and satisfies a certain set of $d-1$ independent inequalities as equations.

For any mixed strategy, the set of pure strategies that have positive probability is called the *support* of this mixed strategy. Two *best response polyhedra* are used to identify the possible supports of strategies in an equilibrium. The best response polyhedron of a player is the set of that player's mixed strategies together with the "upper envelope" of expected payoffs (and larger payoffs) to the other player. Denote by $\overline{P}$ and $\overline{Q}$ the best response polyhedra of player 1 and player 2, given by

$$\overline{P} = \{(x, v) \in \mathbb{R}^\Phi \times \mathbb{R} \mid x \geq \mathbf{0},\ \mathbf{1}^\top x = 1,\ B^\top x \leq \mathbf{1}v\},$$

$$\overline{Q} = \{(y, u) \in \mathbb{R}^\Psi \times \mathbb{R} \mid Ay \leq \mathbf{1}u,\ y \geq \mathbf{0},\ \mathbf{1}^\top y = 1\}.$$

We say a point (x, v) of $\overline{P}$ has *label* i in Φ if $x_i = 0$, and label j in Ψ if $\sum_{i\in\Phi} b_{ij}x_i = v$. Similarly, a point (y, u) of $\overline{Q}$ has label j in Ψ if $y_j = 0$, and label i in Φ if $\sum_{j\in\Psi} a_{ij}y_j = u$.

Assume, without loss of generality, that A and $B^\top$ are nonnegative and have no zero column. We divide each inequality $\sum_{i\in\Phi} b_{ij}x_i \leq v$ by the positive scalar v for $\overline{P}$, and similarly divide each inequality $\sum_{j\in\Psi} a_{ij}y_j \leq u$ by the positive scalar u. The resulting polytopes P and Q are given as

$$P = \{x \in \mathbb{R}^\Phi \mid x \geq \mathbf{0},\ B^\top x \leq \mathbf{1}\},$$

$$Q = \{y \in \mathbb{R}^\Psi \mid Ay \leq \mathbf{1},\ y \geq \mathbf{0}\}.$$

The points in $P - \{\mathbf{0}\}$ and $Q - \{\mathbf{0}\}$ are in one-to-one correspondence with those of $\overline{P}$ and $\overline{Q}$, respectively, via the mappings $x \mapsto (x \cdot v, v)$ and $y \mapsto (y \cdot u, u)$ with $v = 1/\mathbf{1}^\top x$ and $u = 1/\mathbf{1}^\top y$. For each point $(\frac{1}{v})x$ in P, let the label be the same as that of the point (x, v) in $\overline{P}$. Similarly, for each point $(\frac{1}{u})y$ in Q, let the label be the same as that of the point (y, u) in $\overline{Q}$. A vertex pair (x, y) in $P \times Q$ is completely labelled if every label in $\Phi \cup \Psi$ appears as a label of x or y.

This and the following paragraphs of this section review algorithms that are developed and described in Avis et al. (2010). The set of all Nash equilibria is the union of sets of "maximal Nash subsets" (Millham 1974). These maximal Nash subsets are defined by the maximal "cliques", which are the maximal complete bipartite subgraphs of the bipartite graph R on the vertices of $P-\{\mathbf{0}\}$ and $Q-\{\mathbf{0}\}$, with equilibria as edges; Nash equilibria that are pairs of such vertices are also called *extreme equilibria*. The maximal Nash subsets can thus be computed by the *Clique* algorithm with all extreme equilibria as input (see Avis et al. 2010, Algorithm 2).

Algorithm 2.10 (Clique – Equilibrium components) Input: *All pairs* (x,y) *of extreme equilibria.* Output: *All components of Nash equilibria, given as unions of maximal Nash subsets.* Method: *Consider the set of extreme equilibria as a bipartite graph R. Each connected component C of R defines an equilibrium component; enumerate the maximal cliques of C, which define the maximal Nash subsets.*

All extreme equilibria can be found by enumerating all vertices of the best response polytopes and matching the completely labelled pairs:

Algorithm 2.11 (Equilibria by vertex enumeration) Input: *A nondegenerate bimatrix game.* Output: *All Nash equilibria of the game.* Method: *For each vertex x of* $P-\{\mathbf{0}\}$*, and each vertex y of* $Q-\{\mathbf{0}\}$*, if* (x,y) *is completely labelled, output the Nash equilibrium* $(x\cdot 1/\mathbf{1}^\top x, y\cdot 1/\mathbf{1}^\top y)$.

The following *lrsNash* algorithm requires only one of the two vertex enumeration problems to be solved. For a set of labels L, the face of Q defined by having all inequalities corresponding to the elements of L hold has equalities is denoted by $Q(L)$.

Algorithm 2.12 (lrsNash) Input: *Bimatrix game* (A,B). Output: *All extreme equilibria* (x,y). Method: *For each vertex x of* $P-\{\mathbf{0}\}$ *and set L of labels missing from x,*
(a) determine whether $Q(L)$ *is empty or else find a vertex of* $Q(L)$*, and then*
(b) enumerate the vertices y of $Q(L)$ *and output* (x,y).

Audet et al. (2001) introduced the algorithm "Enumeration of Extreme Equilibria" (EEE algorithm). Avis et al. (2010) present some modifications of this EEE algorithm. Both the EEE and the improved EEE algorithm work on bimatrix games. Instead of enumerating all vertices of the best response polyhedra, these algorithms traverse an implicit search tree by depth-first search, forcing one pure strategy to be either zero or a best response at each node of the search tree. If all pure strategies are either best responses or have zero probability, the resulting solution is an equilibrium. The improved EEE algorithm is explained in geometric terms and handles degenerate games significantly better. It has been implemented in Java, first by Rosenberg (2005), using integer arithmetic, which avoids rounding errors.

We extend these algorithms to game trees later in Sections 2.5 and 2.6.

2.4 Equilibria via labelled polyhedra

In this section, we find all the Nash equilibria of a two-player extensive game by using a pair of "reduced polyhedra". These polyhedra for extensive games are analogous to the best response polyhedra $\overline{P}$ and $\overline{Q}$ for bimatrix games. The facets of these polyhedra are labelled. We use the labels to identify the complementary slackness conditions in Theorem 2.7.

Suppose there are s terminal sequences for player 1 and t terminal sequences for player 2. Player 1 has $m-k$ independent terminal sequences that define x_I and $s-(m-k)$ dependent terminal sequences that define x_D. Player 2 has $n-l$ independent terminal sequences that define y_I and $t-(n-l)$ dependent terminal sequences that define y_D. We assign distinct labels to these sequences so that M and S are the sets of labels for the independent and dependent terminal sequences for player 1, and N and T are these sets for player 2, that is,

$$\begin{aligned} x_I &= \{x_i \mid i \in M\}, \quad x_D = \{x_i \mid i \in S\}, \quad u = \{u_i \mid i \in S\}, \\ y_I &= \{y_j \mid j \in N\}, \quad y_D = \{y_j \mid j \in T\}, \quad v = \{v_j \mid j \in T\}. \end{aligned} \tag{2.14}$$

We want to use these labels in order to identify which of the $s+t$ inequalities in (2.9) are binding. One way to define these labels as numbers is in the order of the inequalities in (2.9) and the respective complementarity conditions in (2.10). That is, S,T,M,N are consecutive sets of numbers, with $S=\{1,\dots,s-m+k\}$, $T=\{s-m+k+1,\dots,s-m+k+t-n+l\}$, $M=\{s-m+k+t-n+l+1,\dots,s+t-n+l\}$, $N=\{s+t-n+l+1,\dots,s+t\}$. The particular choice of these labels is not relevant; all that matters is (2.14).

Now consider the following polyhedra, where $\mathbb{R}_{\geq}=\{\alpha\in\mathbb{R}\mid\alpha\geq 0\}$:

$$\begin{aligned} D_1 &= \{(x_I,v)\in\mathbb{R}^M_{\geq}\times\mathbb{R}^T_{\geq}\mid P_2x_I+p_2\geq\mathbf{0},\ -Q_2^\top v-(B')^\top x_I\geq(b')^\top\},\\ D_2 &= \{(y_I,u)\in\mathbb{R}^N_{\geq}\times\mathbb{R}^S_{\geq}\mid Q_2y_I+q_2\geq\mathbf{0},\ -P_2^\top u-A'y_I\geq a'\}. \end{aligned} \tag{2.15}$$

We call D_1 and D_2 "reduced polyhedra" because they are derived from the reduced system (2.9) that has the redundant variables and redundant constraints removed. Each polyhedron is the set of the realization probabilities of the independent sequences together with the vectors of dual variables u or v. Different from the system for bimatrix games, for which the scalar dual variables u and v define the upper envelope of expected payoffs, the upper envelopes of the expected payoffs for two-player extensive games are given by the expressions $c_1+p_2^\top u$ for player 1 and $c_2+q_2^\top v$ for player 2 in (2.7) and (2.8).

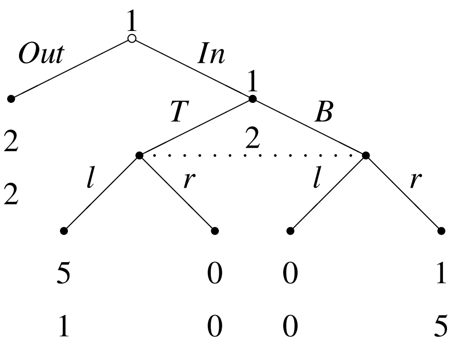
Example 2.13 (Polyhedra for finding Nash equilibria) Consider the extensive game in Example 1.1. The reduced system and a', b', A', B' are shown in Example 2.8. So the polyhedron D_1 is defined by the following system of inequalities:

$$\begin{array}{rrrrrcl} 5 & +4x_{AD} & -4x_{AE} & -4x_{AF} & +v_0 & \geq & 0\\ 3 & & +4x_{AE} & +x_{AF} & +v_0 & \geq & 0\\ 1 & & -x_{AE} & -x_{AF} & & \geq & 0\\ & -x_{AD} & +x_{AE} & +x_{AF} & & \geq & 0\\ & x_{AD}, & x_{AE}, & x_{AF}, & v_0 & \geq & 0 \end{array}$$

and D_2 by the system:

$$\begin{array}{rrrrrcl} & 4y_b & & & +u_1 & \geq & 0\\ 7 & -13y_b & -3y_c & +u_0 & -u_1 & \geq & 0\\ 7 & -13y_b & -11y_c & +u_0 & -u_1 & \geq & 0\\ 1 & -y_b & -y_c & & & \geq & 0\\ & y_b, & y_c, & u_0, & u_1 & \geq & 0. \end{array}$$

The following is a simpler example to illustrate these polyhedra and their labels.



I \ II	∅	l	r
∅			
In			
Out	2, 2		
In, *T*		5, 1	0, 0
In, *B*		0, 0	1, 5

Figure 2.3: Myerson's example of an extensive game, and its sequence form payoff matrix.

Example 2.14 (Myerson 1986) Consider the extensive game by Myerson (1986) and its sequence form payoff matrix shown in Figure 2.3. Let $x_I = (x_{In,T}, x_{In,B})$ and $y_I = (y_r)$; both u and v are here scalars. We have five labels and let in (2.14) $S = \{1\}$, $T = \{2\}$, $M = \{3,4\}$, $N = \{5\}$ (so only M is not a singleton), with $x_I = (x_{In,T}, x_{In,B}) = (x_3, x_4)$ and $y_I = (y_r) = (y_5)$. Then the polyhedra in (2.15) are

$$\begin{aligned} D_1 &= \{(x_I, v) \in \mathbb{R}^{M\times T}_{\geq} \mid -x_3 - x_4 + 1 \geq 0,\ v + x_3 - 5x_4 \geq 0\}, \\ D_2 &= \{(y_I, u) \in \mathbb{R}^{N\times S}_{\geq} \mid u + 5y_5 - 3 \geq 0,\ u - y_5 + 2 \geq 0\}. \end{aligned} \tag{2.16}$$

These polyhedra are shown in Figure 2.4.

In (2.15), the binding inequalities define the *labels* of a point (x_I, v) in D_1 and of a point (y_I, u) in D_2, according to

$$\begin{aligned} L(x_I, v) = {} & \{i \in S \mid (P_2 x_I + p_2)_i = 0\} \cup \{j \in T \mid v_j = 0\} \\ & \cup \{i \in M \mid (x_I)_i = 0)\} \cup \{j \in N \mid (-Q_2^\top v - B' x_I - (b')^\top)_j = 0\}, \\ L(y_I, u) = {} & \{i \in S \mid u_i = 0\} \cup \{j \in T \mid (Q_2 y_I + q_2)_j = 0\} \\ & \cup \{i \in M \mid (-P_2^\top u - A' y_I - a)_i = 0\} \cup \{j \in N \mid (y_I)_j = 0)\}. \end{aligned} \tag{2.17}$$

In Example 2.14, the point $(x_3, x_4, v) = (0,0,0)$ of D_1 has labels $2,3,4,5$, and the point $(y_5, u) = (\frac{3}{5}, 0)$ of D_2 has labels $1,3$, as shown in Figure 2.4.

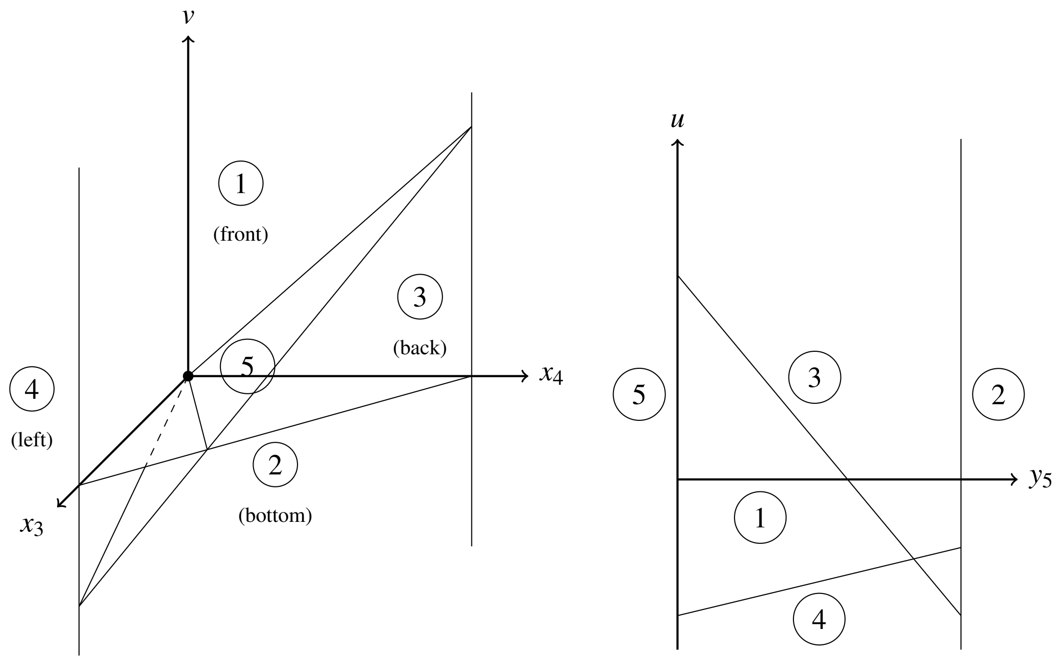


Figure 2.4: Polyhedra D_1 and D_2 for the extensive game of Myerson's Example 2.14. The "bottom" facets of D_1 have labels 2 and 5, those of D_2 have labels 3 and 1.

The following lemmas and theory in this section show that a pair of points in $D_1 \times D_2$ in (2.15) represents a Nash equilibrium if and only if it is *completely labelled*. We prove that such a Nash equilibrium is the convex combination of vertices of the polyhedra. Therefore, all Nash equilibria can be described by Nash equilibria that are vertex pairs of $D_1 \times D_2$.

Here we say a pair of points $((x_I, v), (y_I, u))$ in $D_1 \times D_2$ is *completely labelled* if every label appears as a label of (x_I, v) or (y_I, u), that is, $L(x_I, v) \cup L(y_I, u) = S \cup T \cup M \cup N$.

Let $K, L \subseteq S \cup T \cup M \cup N$ and define

$$\begin{aligned} D_1(K) = \{(x_I, v) \in D_1 \mid & \forall i \in K \cap S : (P_2 x_I + p_2)_i = 0,\ \forall j \in K \cap T : v_j = 0, \\ & \forall i \in K \cap M : x_i = 0,\ \forall j \in K \cap N : (-Q_2^\top v - B' x_I)_j = b'_j\}, \\ D_2(L) = \{(y_I, u) \in D_2 \mid & \forall i \in L \cap S : u_i = 0,\ \forall j \in L \cap T : (Q_2 y_I + q_2)_j = 0, \\ & \forall i \in L \cap M : (-P_2^\top u - A' y_I)_i = a'_i,\ \forall j \in L \cap N : y_j = 0\}. \end{aligned} \tag{2.18}$$

The set of all Nash equilibria is described in the following lemma.

Lemma 2.15 *With (2.4) and (2.18), the pair of realization plans (x, y) is a Nash equilibrium if and only if there are sets K, L and vectors u, v so that $K \cup L = S \cup T \cup M \cup N$ and $((x_I, v), (y_I, u)) \in D_1(K) \times D_2(L)$.*

Proof. Given K and L so that $K \cup L = S \cup T \cup M \cup N$, any $((x_I, v), (y_I, u))$ in $D_1(K) \times D_2(L)$ is completely labelled, so it fulfills the complementarity conditions (2.10), and because $((x_I, v), (y_I, u)) \in D_1 \times D_2$, the constraints in (2.9) hold, so (x, y) is a Nash equilibrium by Theorem 2.7.

Conversely, the constraints (2.9) imply that any Nash equilibrium (x, y) fulfills $((x_I, v), (y_I, u)) \in D_1 \times D_2$. Let $K = L(x_I, v)$ and $L = L(y_I, u)$ according to (2.17), so that $((x_I, v), (y_I, u)) \in D_1(K) \times D_2(L)$. Then $K \cup L = S \cup T \cup M \cup N$ holds because of (2.10). □

By Lemma 2.15, the set of all Nash equilibria is the union of products $D_1(K) \times D_2(L)$ of faces of the polyhedra D_1 and D_2. The following theorem characterizes these products in terms of pairs of vertices or points on extreme rays of D_1 and D_2. We write $\operatorname{conv} U$ for the convex hull of a set U. The following lemma and its proof is analogous to Proposition 4 of Avis et al. (2010).

Lemma 2.16 *Let G be a two-person extensive game, $((x_I, v), (y_I, u)) \in D_1 \times D_2$, and let x and y be defined by (2.4). Then (x, y) is a Nash equilibrium of G if and only if there is a set U of vertices or points on extreme rays of D_1 and a set V of vertices or points on extreme rays of D_2 so that $(x_I, v) \in \operatorname{conv} U$ and $(y_I, u) \in \operatorname{conv} V$, and every $(\mu, \nu) \in U \times V$ is completely labelled.*

Proof. Suppose (x, y) is a Nash equilibrium. By Lemma 2.15, there are sets K and L so that $K \cup L = S \cup T \cup M \cup N$ and vectors u and v so that $((x_I, v), (y_I, u)) \in D_1 \times D_2$. By definition, see (2.18), every point μ in $D_1(K)$ has at least the labels in K, and every ν in $D_2(L)$ has at least the labels in L. Let U and V be the sets of vertices and points on extreme rays of $D_1(K)$ and $D_2(L)$, respectively. Then $D_1(K) = \operatorname{conv} U$ and $D_2(L) = \operatorname{conv} V$ and $((x_I, v), (y_I, u)) \in \operatorname{conv} U \times \operatorname{conv} V$, respectively, and every $(\mu, \nu) \in U \times V$ is completely labelled.

Conversely, given sets of vertices and points on extreme rays U and V so that every $(\mu,\nu) \in U \times V$ is completely labelled, let K be the set of labels common to all $\mu \in U$, and let L be the set of labels common to all $\nu \in V$. Then $K \cup L = S \cup T \cup M \cup N$, because otherwise there would be some label that is missing from some μ in U and from some ν in V, so that (μ,ν) is not completely labelled, contrary to the assumption. Then $(x_I, v) \in \operatorname{conv} U \subseteq D_1(K)$ and $(y_I, u) \in \operatorname{conv} V \subseteq D_2(L)$, and (x,y) is a Nash equilibrium by Lemma 2.15. □

Lemma 2.16 shows that the set of all Nash equilibria can be described by the set of Nash equilibria that are represented by pairs of vertices or points on extreme rays of D_1 and D_2. The Nash equilibria given by pairs of vertices are called *extreme* equilibria in the sense that they are not convex combinations of other equilibria.

For strategic form games, the scalar dual variables u and v always represent equilibrium payoffs and can be easily eliminated. The polyhedra are reduced to a pair of polytopes. Thus all the extreme equilibria are represented by pairs of vertices of the polytopes. For extensive games, since the dual variables u and v are not scalar, the polyhedra cannot be replaced by polytopes. A polyhedron is the convex hull of its vertices and extreme rays. The following lemmas show that every Nash equilibrium that is a convex combination of points on extreme rays is also a convex combination of vertices.

Lemma 2.17 *Suppose (x,y) represents an equilibrium where (x_I, v') is a point in D_1 and (y_I, u') is a point in D_2. Then neither (x_I, v') nor (y_I, u') is a point on an extreme ray, except for the endpoint of such a ray.*

Proof. All points on a ray of D_1 or D_2 fulfill the same tight inequalities and thus have the same labels, except for the endpoint of such a ray which has additional labels. So if (x_I, v') or (y_I, u') is a point on a ray and not an endpoint, all points on that ray have the same labels and would represent parts of an equilibrium, giving vectors (x_I, v') or (y_I, u') with at least one unbounded component. We show that this is not case, which is immediate for the components of x_I or y_I but not for u' or v'. We consider player 1 and show that u' is bounded in any equilibrium; the same consideration for player 2 implies that v' is also bounded.

We proceed in three steps: First, we show that the unreduced system (1.9) has only bounded optimal solutions u. Second, we show how the reduced system where x is expressed as a linear function of x_I according to (2.4) has a dual LP whose solutions are bounded because they are linear functions of the solutions u of the unreduced dual system. Third, we show how these considerations also apply after eliminating redundant inequalities in the reduced primal system, which has fewer dual variables u'.

Let x and y be realization plans. Then x is a best response to y if it solves the primal LP (1.5), which has the dual LP (1.9). The first primal constraints in $Ex = e$ is $x(\emptyset) = 1$; let its corresponding dual variable by u_0. Any other row of $Ex = e$ is given by (2.1) for one of player 1's information sets h in H_1; let its corresponding dual variable by u_h. The dual constraints in $E^\top u \geq Ay$ correspond to the empty sequence $\emptyset$ and to the sequence $\sigma_h c$ for each h in H_1 and c in C_h. They are therefore given by $u_0 - \sum_{h \in H_1 : \sigma_h = \emptyset} u_h \geq (Ay)_\emptyset$ or equivalently

$$u_0 \geq (Ay)_\emptyset + \sum_{h \in H_1 : \sigma_h = \emptyset} u_h \tag{2.19}$$

and correspondingly for the nonempty sequences $\sigma_h c$ by

$$u_h \geq (Ay)_{\sigma_h c} + \sum_{k \in H_1 : \sigma_k = \sigma_h c} u_k \qquad (c \in C_h,\ h \in H_1) \tag{2.20}$$

(see also von Stengel 1996, p. 239). We use (2.19) and (2.20) inductively, starting with the information sets h closest to the leaves, to show that each component u_h or u_0 of u is bounded from below. If $\sigma_h c$ is a terminal sequence, then the sum on the right hand side in (2.20) is empty and u_h is bounded from below because the entries of Ay are bounded. If $\sigma_h c$ is not a terminal sequence, then there exists k in H_1 with $\sigma_k = \sigma_h c$. By induction, we can assume that u_k is bounded from below, so that (2.20) shows that u_h is also bounded. Eventually, (2.19) shows that u_0 is bounded from below. In an optimal solution, (2.19) holds as equality by the complementary slackness condition because the primal variable $x_\emptyset$ is nonzero; also, u_0 is equal to the optimal value $e^\top u$ of the dual LP and thus represents the best response payoff to player 1. So u_0 is bounded from above and we can now use (2.19) and (2.20) with induction down the tree to show that u_h, for information sets h with successively longer incoming sequences σ_h, are bounded from above. This shows u is bounded as claimed. The same reasoning applied to the second player shows that v in an optimal solution to the dual LP (1.10) is also bounded when (x,y) is an equilibrium.

The second step is to consider the reduced system, first without omitting redundant inequalities. Let the row vector a be given by $a = (Ay)^\top$. The primal LP (1.5) says: maximize ax subject to $Ex = e$, $x \ge 0$. Its dual (1.9) says: minimize $u^\top e$ subject to $u^\top E \ge a$ (for readability, we transpose the vector u of variables rather than the constraints). Split the columns of E into sets B and I of dependent and independent variables, respectively, as in (2.2), with $a = (a_B, a_I)$, so that the primal LP says

$$\begin{array}{lccccc} \text{maximize} & a_B x_B & + & a_I x_I & & \\ \text{subject to} & E_B x_B & + & E_I x_I & = & e \\ & x_B & , & x_I & \ge & 0 \end{array} \tag{2.21}$$

and the dual

$$\begin{array}{ll} \text{minimize} & u^\top e \\ \text{subject to} & u^\top E_B \ge a_B, \quad u^\top E_I \ge a_I\,. \end{array} \tag{2.22}$$

The equations in (2.21) are equivalent to $x_B = E_B^{-1}e - E_B^{-1}E_I x_I$, so that if we use this equation to express $x = (x_B, x_I)$ in terms of x_I then the primal LP (2.21) can be written as

$$\begin{array}{lrcl} \text{maximize} & a_B E_B^{-1} e + (a_I - a_B E_B^{-1} E_I) x_I & & \\ \text{subject to} & E_B^{-1} E_I x_I & \le & E_B^{-1} e \\ & x_I & \ge & 0 \end{array} \tag{2.23}$$

where the inequalities stand for $x_B \ge 0$ and the equations $Ex = e$ hold implicitly. This LP has a constant $a_B E_B^{-1} e$ added to its objective function, which is simply added to the usual dual objective function to obtain the dual LP. The dual of (2.23) with variables w is therefore

$$\begin{array}{llccl} \text{minimize} & a_B E_B^{-1} e + & w^\top E_B^{-1} e & & \\ \text{subject to} & & w^\top E_B^{-1} E_I & \ge & a_I - a_B E_B^{-1} E_I \\ & & w & \ge & 0 \end{array} \tag{2.24}$$

which is equivalent to

$$\begin{array}{llll} \text{minimize} & (a_B + w^\top)E_B^{-1}e & & \\ \text{subject to} & (a_B + w^\top)E_B^{-1}E_I & \geq & a_I \\ & w & \geq & 0\,. \end{array} \tag{2.25}$$

Suppose w is feasible and optimal for (2.25), and let $u^\top = (a_B + w^\top)E_B^{-1}$. Then $u^\top E_I \geq a_I$, and $w^\top = u^\top E_B - a_B \geq 0$, that is, $u^\top E_B \geq a_B$, so u is feasible for the unreduced dual LP (2.22). Moreover, $u^\top e = (a_B + w^\top)E_B^{-1}e$, so u is optimal for (2.22) if and only if w is optimal for (2.25) which holds when the dual objective function has the same value as the primal objective function, which is the same for the unreduced and reduced primal LPs (2.21) and (2.23). So w can be unbounded in an optimal solution to (2.25) if and only if u can be unbounded in an optimal solution to (2.22), which we have shown above is not possible. Note here that $a = (Ay)^\top$ is a bounded function of y, which is bounded in the same way if y is expressed in terms of its independent variables y_I due the constraints $Fy = f$, $y \geq 0$ for y.

The third step is to note that in our derivation of the polyhedra D_1 and D_2, not all primal constraints $x_B \geq 0$ which define the inequalities in (2.23) have been used because $x_B = (x_N, x_D)$ and the inequalities $x_N \geq 0$ are redundant (recall that N contains the nonterminal sequences). By (2.3), the inequalities $E_B^{-1}E_I x_I \leq E_B^{-1}e$ in (2.23) are $-P_1 x_N \leq p_1$ and $-P_2 x_D \leq p_2$, where only the latter are irredundant. When omitting the former from (2.23), the corresponding dual system is like (2.24) with $w = (w_N, w_D)$ where w_N is set to zero, that is, it has only dual variables w_D which we call $u' = w_D$ in our consideration of points (x_I, v') and (y_I, u') of D_1 and D_2. Any optimal solution u' to the dual LP (for the reduced primal LP that uses only the irredundant inequalities $-P_2 x_D \leq p_2$) gives rise to an optimal solution $w = (w_N, w_D) = (0, u')$ of (2.24), because the complementary slackness condition holds for the rows indexed by N since $w_N = 0$. So if u' was unbounded, then w would be unbounded, which we have shown impossible. □

The following theorem for finding all Nash equilibria is a corollary of the above lemmas.

Theorem 2.18 *Let G be a two-person extensive game, and* $((x_I,v),(y_I,u)) \in D_1 \times D_2$. *The vectors x and y are defined by (2.4). Then* (x,y) *is a Nash equilibrium of G if and only if there is a set U of vertices of* D_1 *and a set V of vertices of* D_2 *so that* $(x_I,v) \in \operatorname{conv} U$ *and* $(y_I,u) \in \operatorname{conv} V$*, and every* $(\mu,\nu) \in U \times V$ *is completely labelled.*

Theorem 2.18 shows that all Nash equilibria can be completely described by Nash equilibria that are represented by vertex pairs of $D_1 \times D_2$, which are finitely many. For example, the two extreme equilibria (d,f) and (d,g) of the game in Example 2.14 represent a component $\{d\} \times \operatorname{conv}\{f,g\}$ of the set of Nash equilibria.

Consider the bipartite graph R on the vertices of D_1 and D_2 whose edges are the completely labelled vertex pairs $((x_I,v),(y_I,u))$. The maximal complete bipartite subgraphs are the maximal "cliques" of R of the form $U \times V$. They define the sets of Nash equilibria $\operatorname{conv} U \times \operatorname{conv} V$, whose union is the set of all Nash equilibria. These sets are called "maximal Nash subsets". Analogous to the *Clique* algorithm for bimatrix games (see Avis et al. 2010), for two-player extensive games, all components of Nash equilibria are computed by the following algorithm.

Algorithm 2.19 (Clique for extensive games) Input: *All Nash equilibria represented by vertex pair of the polyhedra of a two-player extensive game.* Output: *All components of Nash equilibria of the extensive game, given as unions of maximal Nash subsets.* Method: *Consider the set of extreme equilibria as a bipartite graph R. Each connected component C of R defines an equilibrium component; enumerate the maximal cliques of C, which define the maximal Nash subsets.*

In the next sections, we concentrate on finding all the vertex pairs that represent equilibria, which define the input for the *Clique* algorithm.

2.5 Extreme equilibria using vertex enumeration

In this section we present a straightforward method to generate all vertex pairs representing equilibria by enumerating all vertices of D_1 and D_2, and matching the completely labelled pairs. We also explain a variant of this method.

Algorithm 2.20 (Enumerating and matching vertices of both polyhedra) Input: *Two-person extensive game G.* Output: *All vertex pairs* (x,y) *that represent equilibria.* Method: *Enumerate all vertices* (x_I, v) *of* D_1 *and* (y_I, u) *of* D_2 *in (2.16). For every* $((x_I, v), (y_I, u))$*, check if it is completely labelled. If yes, compute* (x,y) *by (2.4) and check if* (x,y) *has been output already. If not, output* (x,y).

Enumerating the vertices of a polyhedron by linear inequalities is a standard problem in computational geometry (see Ziegler 1995). One method is *lrs* by Avis and Fukuda (1992). Briefly, the reverse search algorithm works as follows. The algorithm starts at a known initial vertex of the polyhedron, and uses a linear objective function which is maximized at the initial vertex. For any vertex of the polyhedron, the simplex algorithm with e.g. the lexicographic pivoting rule (see Chvátal 1983) generates a unique path to the initial vertex at which the objective function is maximized. This defines a directed tree with the initial vertex as its root and of which every vertex of the polyhedron is a node. The algorithm *lrs* explores this directed tree by a depth-first search from the root. That is, the algorithm reverts the simplex steps by considering recursively for each vertex u of the polyhedron the edges to another vertex v such that the simplex algorithm pivots from v to u.

To get an initial vertex, a generic method is to use the initialization method of the simplex algorithm for getting a feasible basis for the system (see Chvátal 1983). For an LP

$$\begin{array}{lrcll} \text{maximize} & \sum_{j=1}^{n} c_j x_j & & & \\ \text{subject to} & \sum_{j=1}^{n} a_{ij} x_j & \leq & b_i & (i = 1, 2, \ldots, m) \\ & x_j & \geq & 0 & (j = 1, 2, \ldots, n), \end{array}$$

this method computes an initial feasible solution (if any) of the system by solving the *auxiliary problem*

$$\begin{array}{lrcll} \text{maximize} & x_0 & & & \\ \text{subject to} & \sum_{j=1}^{n} a_{ij} x_j - x_0 & \leq & b_i & (i = 1, 2, \ldots, m) \\ & x_j & \geq & 0 & (j = 0, 1, \ldots, n). \end{array}$$

Obviously $x_j = 0$ for all $j = 1, \ldots, m$ and $x_0 = max_i\{|b_i|\}$ is an initial feasible solution for the auxiliary problem. The original LP has a feasible solution if and only if the optimum value of the auxiliary problem is zero, and then the optimum solution for the auxiliary problem is a feasible solution for the original LP.

An alternative to this generic method is to use the structure of D_1 and D_2 and start with realization plans x and y that represent pure strategies, where every sequence has probability zero or one. In order to find corresponding dual variables in (2.7) and (2.8), one can assign them in analogy to (2.19) and (2.20) by starting from the information sets h closest to the leaves and successively assign u_h so as to fulfill all inequalities (2.20) and one of them tightly. This, translated to the reduced systems (2.7) and (2.8), is also easily seen to produce initial vertices for D_1 and D_2.

For the implementation of the algorithm, the technique of *integer pivoting* can be used to guarantee exact solutions and avoid the relatively costly computation of greatest common divisors when computing with fractions of integers; for details see Avis et al. (2010).

For the example game in Example 1.1, *lrs* outputs all five vertices of D_1:

$$(x_{AD}, x_{AE}, x_{AF}, v_0) = (0,0,0,0)$$

$$(x_{AD}, x_{AE}, x_{AF}, v_0) = (0,0,1,0)$$

$$(x_{AD}, x_{AE}, x_{AF}, v_0) = (0,1,0,0)$$

$$(x_{AD}, x_{AE}, x_{AF}, v_0) = (1,0,1,0)$$

$$(x_{AD}, x_{AE}, x_{AF}, v_0) = (1,1,0,0)$$

and all six vertices of D_2:

$$(y_b, y_c, u_0, u_1) = (0,0,0,0)$$

$$(y_b, y_c, u_0, u_1) = (0,0,0,7)$$

$$(y_b, y_c, u_0, u_1) = (0,1,4,0)$$

$$(y_b, y_c, u_0, u_1) = (0, \frac{7}{11}, 0, 0)$$

$$(y_b, y_c, u_0, u_1) = (1,0,6,0)$$

$$(y_b, y_c, u_0, u_1) = (\frac{7}{13}, 0, 0, 0)$$

For the game tree of Figure 1, the only pair generated by Algorithm 2.20 is

$$((x_{AD}, x_{AE}, x_{AF}, v_0), (y_b, y_c, u_0, u_1)) = ((0,0,0,0), (0,0,0)).$$

So the only sequence form equilibrium of the game is

$$((\emptyset, x_A, x_B, x_{AC}, x_{AD}, x_{AE}, x_{AF}), (\emptyset, y_a, y_b, y_c)) = ((1,0,1,,0,0,0,0), (1,1,0,0)).$$

Avis (2011) implemented the *lrsNash* algorithm described in Avis et al. (2010) as the method `nash` to enumerate all extreme equilibria of bimatrix games. Compared to the straightforward algorithm of enumerating all vertices of both polytopes and output all completely labelled pairs, the *lrsNash* algorithm only completely solves one of the two vertex enumeration problems, requires only memory proportional to the input size because only the vertices of one polyhedron are generated sequentially, and does not require a separate matching process. The following *lrsNash-extensive* algorithm for two-person extensive games, analogously, enumerates the vertices (x_I, v) of only one polyhedron D_1. The set $L = (M \cup N \cup S \cup T) - K$ of missing labels from (x_I, v) defines the face $D_2(L)$ of D_2 according to (2.18). If $D_2(L)$ is not empty, then it has a vertex that is be used as a starting point for enumerating all its vertices with *lrs*.

Algorithm 2.21 (lrsNash-extensive) Input: *Two-person extensive game G.* Output: *All extreme equilibria* (x, y). Method: *For each vertex* (x_I, v) *of* D_1 *and set L of labels missing from* (x_I, v)*,*
(a) determine whether $D_2(L)$ *is empty or else find a vertex of* $D_2(L)$*,*
(b) enumerate the vertices (y_I, u) *of* $D_2(L)$*,*
(c) compute (x, y) *according to (2.4), and then*
(d) check if (x, y) *has been output already and if not output* (x, y).

2.6 The modified EEE algorithm for two-player extensive games

Audet et al. (2001) defined the algorithm "Enumeration of all Extreme Equilibria of bimatrix games" (EEE). Avis et al. (2010) have improved that algorithm and called it the "modified EEE algorithm", using arbitrary precision arithmetic and more efficient treatment of degenerate games. Both algorithms work on bimatrix games. Instead of enumerating all vertices of the best response polyhedra of both players they traverse an implicit search tree by depth-first search, and force one pure strategy to be either zero or a best response at each node of the search tree. If all pure strategies are either best responses or have zero probability, the resulting solution is an equilibrium. The improved EEE algorithm is explained in geometric terms and handles degenerate games significantly better. It has been implemented in Java, first by Rosenberg (2005), using integer arithmetic which avoids the rounding errors of the floating-point arithmetic of Audet et al. (2001).

We extend the improved EEE algorithm, and compute all extreme equilibria for two-player extensive games. Like the algorithm for bimatrix games, the algorithm traverses the search tree by depth-first search. Each node of the tree corresponds to a pair of vertices of the polyhedra D_1 and D_2; recall that they have dimension s and t, respectively, because they are defined only by inequalities in (2.15) and are full-dimensional. Given $(K,L,(x_I,v),(y_I,u))$ representing a node of the search tree so that $|K\cup L| < s+t$, a new label h that is not in $K\cup L$ is selected and added to either K or L, which defines the two new branches of the tree. If the resulting subset $D_1(K\cup\{h\})$ or $D_2(L\cup\{h\})$ is empty and thus has no vertex, then the branch is omitted and the search tree pruned at that point.

The algorithm starts at the node of tree which is given by $K=L=\emptyset$ and a pair of vertices $((x_I,v),(y_I,u))$ of $D_1\times D_2$. Set the depth of this node to be zero, and the depth of any other node as one more than the depth of its parent. At depth $s+t$, we have $K\cup L = M\cup S\cup N\cup T$ and at this point the pair (x,y) of this node defines an equilibrium.

Now we apply the EEE algorithm with D_1, D_2, $D_1(K)$ and $D_2(L)$ defined as in (2.18). In the initial step, a pair of vertices is selected. Analogous to Audet et al. (2001), these vertices can be computed as follows. Set $x_I{}'$ to be an arbitrary feasible value, for example, $x_I{}' = (\frac{1}{m-k},\ldots,\frac{1}{m-k})$. Choose (y_I,u) that is feasible for D_2 and maximizes $(b' + x_I^\top B')y_I - p_2^\top u$, and (x_I, u) that is feasible for D_1 and maximizes $x_I^\top(a' + A'y_I^\top) - q_2^\top v$. This produces a vertex pair of $D_1 \times D_2$.

For choosing the label h, we let

$$\tau_i = \begin{cases} (x_I)_i(-P_2^\top u - A'y_I - a')_i & \text{if the variable } (x_I)_i \text{ of } D_1 \text{ is not forced to be} \\ & \text{null, and the inequality } (-P_2^\top u - A'y_I)_i \geq a'_i \\ & \text{of } D_2 \text{ is not fixed at equality,} \\ -1 & \text{otherwise,} \end{cases}$$

$$\pi_j = \begin{cases} u_j(P_2 x_I + p_2)_j & \text{if the variable } u_j \text{ of } D_2 \text{ is not forced to be null,} \\ & \text{and } (P_2 x_I + p_2)_j \geq 0 \text{ of } D_1 \text{ is not fixed at equality,} \\ -1 & \text{otherwise,} \end{cases}$$

$$\varphi_c = \begin{cases} (y_I)_c(-Q_2^\top v - B'x_I - b')_c & \text{if the variable } (y_I)_c \text{ of } D_2 \text{ is not forced to be null,} \\ & \text{and the inequality } (-Q_2^\top v - (B')^\top x_I)_c \geq (b')_c^\top \\ & \text{of } D_1 \text{ is not fixed at equality,} \\ -1 & \text{otherwise,} \end{cases}$$

$$\psi_d = \begin{cases} v_d(Q_2 y_I + q_2)_d & \text{if the variable } v_d \text{ of } D_1 \text{ is not forced to be null,} \\ & \text{and } (Q_2 y_I + q_2)_d \geq 0 \text{ of } D_2 \text{ is not fixed at equality,} \\ -1 & \text{otherwise.} \end{cases}$$

Select the index h among i, j, c, d that gives the maximum of τ_i, π_j, φ_c and ψ_d. That is, the label h is chosen so that $(x_I)_h(-P_2^\top u - A'y_I - a')_h$ or $u_h(P_2 x_I + p_2)_h$ or $(y_I)_h(-Q_2^\top v - B'x_I - b')_h$ or $v_h(Q_2 y_I + q_2)_h$ is maximal among these products, with smallest h in case of ties.

The algorithm is summarized as follows, in analogy to Algorithm *EEE-m* of Avis et al. (2010):

Algorithm 2.22 (EEE for extensive games) Input: *Two-person extensive game G.* Output: *All extreme equilibria* (x, y). Method: *Implicit depth-first search on a binary tree by choosing any vertices* (x_I, v) *of* D_1 *and* (y_I, u) *of* D_2*, and calling* `visit-extensive(∅,∅,`x_I, y_I`)` *with the recursive* `visit-extensive` *method. For each* (x_I, y_I) *output by* `visit-extensive`*, compute* (x, y) *from* (x_I, y_I) *by (2.4). Output* (x, y) *if not already output earlier.*

The `visit-extensive` method is a standard recursive depth-first exploration of a search tree:

`visit-extensive`(K, L, x_I, y_I):
[*assumption:* (x_I, v) *vertex of* $D_1(K)$, (y_I, u) *vertex of* $D_2(L)$]
if $|K \cup L| < m+n+s+t$ **then**
  select $h \in (M \cup N \cup S \cup T) - (K \cup L)$
  if $|K| < m+s$ and $\exists$ vertex $((x_I)', v')$ of $D_1(K \cup \{h\})$ **then**
    `visit`$(K \cup \{h\}, L, (x_I)', y_I)$
  if $|L| < n+t$ and $\exists$ vertex $((y_I)', u')$ of $D_2(L \cup \{h\})$ **then**
    `visit`$(K, L \cup \{h\}, x_I, (y_I)')$
else
  for all vertices (x_I, v) of $D_1(K)$ and (y_I, u) of $D_2(L)$ **do**
    output (x_I, y_I) if not already output earlier.

As an example, consider the extensive game in Example 2.14. The tree shown in Figure 2.5 displays the complete search tree generated by the algorithm. The circled numbers indicate the order in which the nodes are selected.

We start with the vertex $(x_1, x_2, v_5) = (0,0,0)$ in $D_1(\emptyset)$ and $(y_4, u_3) = (0,3)$ in $D_2(\emptyset)$. The first extreme equilibrium is found at node 15, with $x_1 = x_2 = v_5 = 0$, $K = \{1,2,4,5\}$ and $L = \{3\}$. Two extreme equilibria are found by the two vertices $(y_4, u_3) = (1,0)$ and $(y_4, u_3) = (\frac{3}{5}, 0)$ in $D_2(L)$. The same extreme equilibria are found at node 22 but are not recorded again. The extreme equilibrium with $(x_1, x_2, v_5) = (0,0,0)$ and $(y_4, u_3) = (1,0)$ is also found at node 16 but is not recorded again. Thus the algorithm outputs three extreme equilibria:

$$(x_\emptyset, x_{In}, x_{Out}, x_{In,T}, x_{In,B}) = (1,0,1,0,0), (y_\emptyset, y_l, y_r) = (1,0,1),$$

and

$$(x_\emptyset, x_{In}, x_{Out}, x_{In,T}, x_{In,B}) = (1,0,1,0,0), (y_\emptyset, y_l, y_r) = (1, \frac{2}{5}, \frac{3}{5}).$$

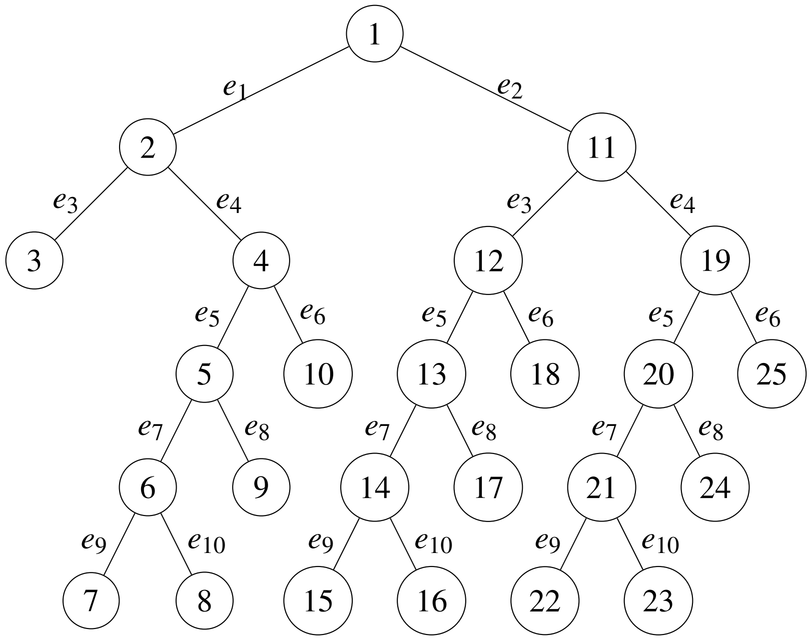


Figure 2.5: Search tree generated by EEE algorithm. The equations represented by $e_1, \cdots, e_{10}$ are: $e_1 : -x_1 - x_2 + 1 = 0$, $e_2 : u_3 = 0$, $e_3 : x_1 = 1$, $e_4 : u_3 + y_5 - 3 = 0$, $e_5 : x_2 = 0$, $e_6 : u_3 - y_4 + 2 = 0$, $e_7 : v_5 + x_1 - 5x_2 = 0$, $e_8 : y_4 = 0$, $e_9 : v_5 = 0$, and $e_{10} : -y_4 + 1 = 0$.

CHAPTER 3

Computing an Extensive Form Correlated Equilibrium

3.1 Introduction

In this chapter we focus on the computation of one Extensive Form Correlated Equilibrium (EFCE). We prove that an EFCE can be computed by a polynomial time algorithm. This algorithm is achieved by applying analogous steps of the algorithm (Papadimitriou and Roughgarden 2008, Jiang and Leyton-Brown 2010) for computing CE to a system of EFCE that is similar to the EFCE description by sequences (von Stengel and Forges 2008). At the time of this writing, there are ongoing discussions about some necessary amendments of the algorithm by Papadimitriou and Roughgarden, which can be similarly applied to our adaption of this method; this is discussed further in Section 3.3.

The existence of Nash equilibrium in any game is well known. In strategic form games, while the complexity of computing Nash equilibrium is PPAD-complete even for two-player games (Chen and Deng 2006, Daskalakis, Goldberg and Papadimitriou 2006), the computation problem for CE is typically "easier", that is, polynomial-time solvable (Gilboa and Zemel 1989). In addition, a correlated equilibrium for a succinctly represented game is easier to compute, in the sense that the complexity can be polynomial, even for multi-player games (Papadimitriou and Roughgarden 2008).

For extensive games, there is not an easy way to get a polynomial time algorithm by transforming the game tree into normal form and applying the algorithm for CE directly to it. This holds because a player may have an exponential number of strategies, so that the length of transformed normal form itself is exponential in the size of the game tree input. In fact, the set of strategic-form CE cannot be a polytope defined by polynomially many inequalities, unless $P = NP$, because finding a max-payoff CE is NP-hard (von Stengel 2001, Chu and Halpern 2001).

This motivates a different concept, the EFCE, where a player only gets move recommendations, and so has to compare moves at each information set, instead of comparing strategies. The idea of this sort of correlated equilibrium was first proposed by Myerson (1986), and elaborated by von Stengel and Forges (2008). As mentioned in Chapter 1, von Stengel and Forges proposed a compact description for EFCE of two-player games without chance moves, and proved that the complexity of computing all EFCE of two-player games without chance moves is polynomial. However, the consistency and incentive constraints in their compact description are only necessary but not sufficient conditions for multi-player games or when chance moves are introduced. In fact, computing a maximum-payoff EFCE for multi-player games or two-player games with chance moves is NP-hard (von Stengel and Forges 2008).

Papadimitriou and Roughgarden's polynomial time algorithm for computing a CE is based on the proof of the existence of a CE that uses LP duality (Hart and Schmeidler 1989, Nau and McCardle 1990). Briefly speaking, this proof exploits the existence of stationary distributions for Markov chains to show the infeasibility of the dual problem of an LP, whose normalized nontrivial solution represents a CE. By the duality theorem, this LP is unbounded and thus a CE exists. This proof provides an interpretation of the dual system. In this interpretation, the dual variables are considered as what we call the players' "deviation plan", and the inequalities describe a rationality concept called "joint coherence" (Nau and McCardle 1990).

This polynomial time algorithm can be applied to a succinctly specified game, i.e. a game with polynomial number of players, the polynomial expectation property for product distributions, and of polynomial type (meaning a polynomial number of actions per player). When looking at behaviour strategies for extensive games, we find that all the conditions are satisfied except for the polynomial type: a player has an exponential number of strategies to deviate to when given a recommendation. However, this is potentially recoverable because given a fixed behaviour of the other players, a best response, even from a set of exponentially many pure strategies, can be computed in polynomial time. This is essentially done by backward induction, as already described by Wilson (1972), and is used by Koller and Megiddo (1992) in a precursor to the sequence form. This can also be captured by the system of incentive constraints described above when considering the EFCE concept.

The goal of this chapter is to adapt Papadimitriou and Roughgarden's algorithm to compute an EFCE. In Section 3.4, we construct the system for EFCE using product profiles rather than sequences. We provide the proof for the existence of EFCE and the algorithm for EFCE computation. The main theorem in this chapter is the following; its contribution is its constructive proof with the help of LP duality that can be turned into a polynomial-time algorithm.

Theorem 3.1 (Existence of EFCE, Huang and von Stengel 2008) *Every multi-player, perfect-recall extensive game has an EFCE, which can be constructed with the help of LP duality.*

In our EFCE description using product profiles, given a recommended move at an information set, the incentive constraints compare the *expected payoff contribution* that the player gets by following the recommendation, with the *optimal expected payoff* that the player obtains by deviating. We prove that an EFCE is a correlated device such that the expected payoff contribution is no less than the optimal expected payoff for every move of every player.

This system is suitable also for computational considerations. As mentioned earlier, all conditions for a succinctly specified game are either satisfied or recoverable. Thus it is possible to compute a EFCE in polynomial time with this system.

The existence proof for EFCE is adapted from that for CE. Induction is used based on the structure of the tree. Details are provided in Section 3.5.

Recently there have been discussions on the correctness of Papadimitriou and Roughgarden's algorithm. Stein, Parrilo and Ozdaglar (2010) claimed to have found an error in this algorithm. In a recent online comment, Papadimitriou (2010) expressed confidence that the algorithm could still achieve polynomial time CE identification without dramatic changes. Stein et al. have withdrawn their paper from arXiv. The latest update on this discussion is that Jiang and Leyton-Brown (2010) have presented an extension of the algorithm that avoids the numerical problems and computes a CE in polynomial time. In Section 3.6, we show that both the original and the extension by Jiang and Leyton-Brown of the algorithm for CE computation can be adapted to EFCE. Thus an EFCE can also be computed in polynomial time, as stated in Theorem 1.14.

This chapter is joint work with Bernhard von Stengel. A version of this chapter has been presented in WINE 2008 and published in Lecture Notes in Computer Science (Huang and von Stengel 2008).

3.2 Background

In this section we review the proof of CE existence, and the description for EFCE of two-player games without chance moves.

As mentioned in the previous section, generally the CE concept has better computational complexity properties than Nash Equilibrium (NE). Gilboa and Zemel (1989) summarize the complexity results of some computational problems for both NE and CE for strategic form games. They conclude that, broadly speaking, NE is a complicated solution concept whereas CE is a simple one.

Papadimitriou and Roughgarden propose a polynomial time algorithm to compute one CE for a succinctly specified game. The three conditions for a succinctly specified game are:

1. it has a polynomial number of players;
2. it has the polynomial expectation property for product distributions; and
3. it is of polynomial type.

Their algorithm is based on a constructive proof of CE existence that uses LP duality (Hart and Schmeidler 1989, Nau and McCardle 1990). The proof uses the following lemma, adapted from the Lemma by Hart and Schmeidler (1989).

Lemma 3.2 *Let Y_{ij} be nonnegative numbers for $1 \leq i, j \leq n$. Then there are probabilities $x_1, \ldots, x_n$ so that for all i*

$$x_i \sum_{j=1}^{n} Y_{ij} = \sum_{k=1}^{n} x_k Y_{ki} \tag{3.1}$$

where if $\sum_{j=1}^{n} Y_{rj} = 0$ for some r, then $x_r = 1$ and $x_k = 0$ for $k \neq r$, and otherwise for $1 \leq i \leq n$

$$x'_i = \frac{u_i}{\sum_{j=1}^{n} Y_{ij}}, \qquad x_i = \frac{x'_i}{\sum_{j=1}^{n} x'_j} \tag{3.2}$$

with $(u_1, \ldots, u_n)$ as a stationary distribution of the Markov chain that moves from i to j with probability $Y_{ij} / \sum_{k=1}^{n} Y_{ik}$.

Proof. If $\sum_{j=1}^{n} Y_{rj} = 0$ for some r, then with $x_r = 1$ and $x_k = 0$ for $k \neq r$ both sides of (3.1) are always zero. Otherwise, let $(u_1, \ldots, u_n)$ be a stationary distribution of the described Markov chain, that is, for all i we have

$$u_i = \sum_{k=1}^{n} u_k \frac{Y_{ki}}{\sum_{j=1}^{n} Y_{kj}}$$

which with x'_i as in (3.2) and its constant multiple x_i is exactly (3.1) as claimed. □

An example for the numbers in Lemma 3.2 is $(Y_{ij}) = \begin{bmatrix} 1 & 1 \\ 9 & 1 \end{bmatrix}$, with Markov chain $\begin{bmatrix} 0.5 & 0.5 \\ 0.9 & 0.1 \end{bmatrix}$, which has the stationary distribution $(u_1, u_2) = (\frac{9}{14}, \frac{5}{14})$, so that

$$(x'_1, x'_2) = (\frac{9}{14 \cdot 2}, \frac{5}{14 \cdot 10}), \qquad (x_1, x_2) = (0.9, 0.1).$$

Each probability u_i of the stationary distribution of the Markov chain has to be rescaled as in (3.2), which has been overlooked in Papadimitriou and Roughgarden (2008, p. 14:9).

This lemma is used to prove the existence of a CE.

Theorem 3.3 *Every game has a correlated equilibrium.*

Proof. Consider the LP

$$\begin{aligned} \text{maximize} \quad & \textstyle\sum x_s \\ \text{subject to} \quad & Ax \geq 0 \\ & x \geq 0 \end{aligned} \tag{3.3}$$

where the constraints $Ax \geq 0$ are the matrix form of (1.13) in the definition of CE in Chapter 1. This LP is either trivial, with maximum 0, or unbounded. There exists a CE if and only if it is unbounded. By the LP duality theorem, we can show the LP being unbounded by showing the infeasibility of its dual, which is

$$\begin{aligned} A^\top y \quad & \leq -1 \\ y \quad & \geq 0, \end{aligned} \tag{3.4}$$

This can be achieved by showing that for every $y \geq 0$ there is a product distribution x such that $x^\top A^\top y = 0$. If (3.4) is feasible then there exists y such that all components of $A^\top y$ are negative, which is a contradiction to its convex combination $x^\top A^\top y$ being zero. Let $x : S \to \mathbb{R}$ be a product distribution with $x(s) = \prod_{p \in P} x^p(s_p)$. Then (summing over all players p in P and strategies i, j, k in S_p and s_{-p} in S_{-p})

$$\begin{aligned} x^\top A^\top y &= \sum_{p,i,j} \sum_{s_{-p}} x(i, s_{-p}) \, [a^p(i, s_{-p}) - a^p(j, s_{-p})] \, y^p_{ij} \\ &= \sum_{p,i,j} \sum_{s_{-p}} x(i, s_{-p}) \, a^p(i, s_{-p}) \, y^p_{ij} - \sum_{p,i,j} \sum_{s_{-p}} x(i, s_{-p}) \, a^p(j, s_{-p}) \, y^p_{ij} \\ &= \sum_{p,i,j} \sum_{s_{-p}} x(i, s_{-p}) \, a^p(i, s_{-p}) \, y^p_{ij} - \sum_{p,k,i} \sum_{s_{-p}} x(k, s_{-p}) \, a^p(i, s_{-p}) \, y^p_{ki} \\ &= \sum_{p,i} \sum_{s_{-p}} a^p(i, s_{-p}) \prod_{q \neq p} x^q(s_q) \left(x^p(i) \sum_j y^p_{ij} - \sum_k x^p(k) \, y^p_{ki} \right). \end{aligned}$$

Let $Y_{ij} = y^p_{ij}$ and choose $x^p(i)$ like x_i in Lemma 3.2. This sets the last term in parentheses to zero, which shows that $x^\top A^\top y = 0$ as claimed. □

Hart and Schmeidler (1989) give the above proof of the existence of a CE by introducing a zero-sum auxiliary game. This proof inspires an analogous proof of existence of an EFCE. In the auxiliary game in their proof, player I chooses a strategy profile s and player II chooses a triple p, i, j where $p \in P$ and $i, j \in S_p$. Then if $i = s_p$, player II pays to player I the amount $a^p(i,s_{-p}) - a^p(j,s_{-p})$, otherwise player II pays zero.

Suppose player II employs a mixed strategy y which chooses the triple p,i,j with probability y^p_{ij}. Then the expected payment from player II to player I if player I chooses s is $(A^\top y)_s$, because the payoff matrix to player I is A when player I is the column player.

It suffices to show that for any action of player II, a product distribution x exists to "neutralize" player II's expected payoffs in the sense that $x^\top A^\top y = 0$. It is useful consider the payment $a^p(i,s_{-p}) - a^p(j,s_{-p})$ as split into a first payment $a^p(i,s_{-p})$ from player II to player I and a second payment $a^p(j,s_{-p})$ from player I to player II. The probability of player II paying $a^p(i,s_{-p})$ to player I is given by $\prod_{q \neq p} x^q(s_q) x^p(i) \sum_j y^p_{ij}$, and the probability of player I paying $a^p(i,s_{-p})$ to player II is $\prod_{q \neq p} x^q(s_q) \sum_k x^p(k) y^p_{ki}$. Then we can use Lemma 3.2 to show that there exists x with

$$\begin{aligned}
& \prod_{q \neq p} x^q(s_q) x^p(i) \sum_j y^p_{ij} - \prod_{q \neq p} x^q(s_q) \sum_k x^p(k) y^p_{ki} \\
= & \prod_{q \neq p} x^q(s_q) \left[x^p(i) \sum_j y^p_{ij} - \sum_k x^p(k) y^p_{ki} \right] \qquad (3.5) \\
= & \; 0.
\end{aligned}$$

Note that y is a mixed strategy of player II only if $\sum y^p_{ij} = 1$. Because the dual variables y^p_{ij} do not necessarily sum to 1, they have to be re-scaled in the CE existence proof that uses an auxiliary game. However, this does not affect the contradiction obtained from using a neutralizing strategy x. The only case where it may fail is when $\sum y^p_{ij} = 0$, when $A^\top y \leq -1$ is not fulfilled anyway.

Myerson (1997) used this interpretation to obtain further properties of the strategic CE concept. He also modified the LP and added some variables so that in the dual system $\sum_j y^p_{ij} = 1$ holds for each p,i and a re-scaling of the probabilities of a stationary distribution as in Lemma 3.2 is not necessary.

The goal of this chapter is to adapt the above technique in the proof of CE existence, and the algorithm of CE computation, to EFCE. We first review the description of EFCE for two-player games by von Stengel and Forges (2008). As mentioned earlier, this description is not suitable for multi-player games or games with chance moves. However, the polynomial-sized *incentive constraints* proposed in it can be adapted to our computation of an EFCE.

For the simple case of two-player, perfect recall extensive game without chance moves, von Stengel and Forges (2008) prove that the set of EFCE can be described with polynomially many constraints. In such games, the EFCE can be characterized by the concept of a *correlation plan* based on sequences. Let ξ_1 and ξ_2 be the set of sequences of player 1 and 2, respectively.

Theorem 3.4 (von Stengel and Forges 2008) *In a two-player, perfect-recall extensive game without chance moves, an EFCE is a correlation device induced by a function* $z : \xi_1 \times \xi_2 \rightarrow \mathbb{R}$ *that fulfills the following consistency constraints (i), (ii), for all* $\sigma \in \xi_1$, $h \in H_1$ *and* $\tau \in \xi_2$, $k \in H_2$ *such that there is a path from the root to a leaf containing a node of h and a node of k:*

$$(i)\ \sum_{c \in C_h} z(\sigma_h c, \tau) = z(\sigma_h, \tau), \quad \sum_{d \in C_k} z(\sigma, \tau_k d) = z(\sigma, \tau_k),$$

$$(ii)\ z(\emptyset, \emptyset) = 1, \quad z(\sigma, \tau) \geq 0,$$

and the incentive constraints for any information sets h and k of player 1*, with* $k = h$ *or h preceding k, and* $\sigma \in \xi_1, \sigma = \sigma_h c$*:*

$$(iii)\ u(\sigma) = \sum_{\tau} z(\sigma, \tau) a(\sigma, \tau) + \sum_{k \in H_1 | \sigma_k = \sigma} \sum_{d \in C_k} u(\sigma_k d),$$

$$(iv)\ v(k, \sigma) \geq \sum_{\tau} z(\sigma, \tau) a(\sigma_k d, \tau) + \sum_{l \in H_1 | \sigma_l = \sigma_k d} v(l, \sigma) \quad \text{for } d \in C_k \text{, and}$$

$$(v)\ v(h, \sigma_h c) = u(\sigma_h c),$$

where $a(\sigma, \tau)$ *is the payoff to player* 1*, and similarly for information sets and sequences of player* 2.

A function z that fulfills the consistency constraints (i) and (ii) is called a *correlation plan*.

For every sequence σ, the auxiliary variable $u(\sigma)$ is the expected payoff that player 1 gets when he reaches information set h and follows the recommended move c. According to (iii), this is obtained from payoff contributions at leaves given by $z(\sigma,\tau)a(\sigma,\tau)$ with the entries $a(\sigma,\tau)$ of the sparse sequence form payoff matrix, plus contributions $u(\sigma_k d)$ from longer sequences $\sigma_k d$. For the incentive constraints (iv), notice that the player's posterior on the behavior of the other player is given by the correlation plan and the recommendation he gets at the information set h where where $\sigma = \sigma_h c$. If the player deviates after he gets to information set h and considers another move at this information set or at a subsequent information set, the posterior of this player does not change but remains based on the last recommendation he gets. Therefore $v(k,\sigma)$ is the optimal expected payoff to player 1 if at information set h he gets the recommended move c, and considers another move at this information set or at a subsequent information set k. When the incentive constraints are satisfied, the expected payoffs by following the recommendations are no less than the optimal expected payoffs; hence, the players do not have an incentive to deviate.

These consistency and incentive constraints define a polynomial description of the set of EFCE. This is possible because in a two-player game without chance moves, for every correlation plan that fulfills the consistency constraints, there is a probability distribution μ on the set of strategy profiles S such that for each sequence pair (σ,τ),

$$z(\sigma,\tau) = \sum_{\substack{(p_1,p_2)\in S \\ (p_1,p_2) \text{ agrees with } (\sigma,\tau)}} \mu(p_1,p_2)$$

where (p_1,p_2) *agrees with* (σ,τ) if p_1 chooses all the moves in σ and p_2 chooses all the moves in τ. That is, for every z found by the system of consistency and incentive constraints, a unique correlation device can be defined. Von Stengel and Forges (2008) explain how the required correlation device is induced by generating move recommendations from the correlation plan. Briefly, the generation of moves starts from the root of the game tree. For the moves of an information set of player p, consider a "reference sequence" of the other player that leads to this information set. Because the game has only two players and no chance moves, for every selected reference sequence, the information set is uniquely determined by the player's own history path and the reference sequence.

This description is not suitable for multi-player games or games with chance moves. Von Stengel and Forges (2008) show, by an example, that for such games, the consistency constraints (i) and (ii) in Theorem 3.4 do not suffice to characterize the convex hull of pure strategy profiles. That is, there may be a distribution on sequence pairs that fulfills (i) and (ii) but is not a convex combination of pure strategy pairs (and thus cannot represent an EFCE).

3.3 Recent developments on CE computation

In this section, we review Papadimitriou and Roughgarden's (2008) polynomial time algorithm for computing CE. We summarize recent developments on this algorithm, including Stein, Parrilo, and Ozdaglar's (2010) claim on the error they found in this algorithm, and Jiang and Leyton-Brown's (2010) variant that they called "the Simplified Ellipsoid Against Hope" algorithm.

Based on the constructive proof of CE existence described in the previous section, Papadimitriou and Roughgarden describe a polynomial time algorithm for CE computation that they call the "Ellipsoid Against Hope" algorithm. This algorithm runs the ellipsoid algorithm on the dual LP (3.4). (A review of the ellipsoid algorithm is given in the Appendix.) According to the proof of existence, at each step i of the ellipsoid algorithm, for the candidate solution $y^{(i)}$ there is a product distribution $x^{(i)}$ such that $(x^{(i)}A^\top)y \leq -1$ is violated by $y = y^{(i)}$. This is used as the separation oracle. The ellipsoid algorithm will stop after a polynomial number of steps and determines that the program is infeasible. Let $X = (x^{(i)})_i^\top$ be the matrix whose rows i are the generated product distributions. Consider the LP

$$[XU^\top]y \leq -1,\ y \geq 0. \tag{3.6}$$

If we apply the same ellipsoid algorithm to (3.6), the algorithm will go through the same sequence of queries $y^{(i)}$ and cutting planes that violate $x^{(i)}U^\top y \leq -1$ and return infeasibility. This implies its dual

$$[UX^{\top}]\alpha \geq 0,\ \alpha \geq 0 \tag{3.7}$$

is unbounded. Note that the entries of $UX^{\top}$ are differences between expected payoffs under different product distributions, and so can be computed in polynomial time for succinctly specified games. Also, because (3.6) is of polynomial size, a nonzero feasible α can be computed in polynomial time. We can scale α to obtain a probability distribution. Then the convex combination $X^{\top}\alpha$ of the product distributions $x^{(i)}$ is a CE, as it is feasible for (3.3).

In the paper that they have recently withdrawn, Stein, Parrilo and Ozdaglar (2010) raised concerns about the Ellipsoid Against Hope algorithm. They showed a counterexample in which the algorithm failed to compute a CE. They pointed out that the output of this algorithm satisfied all of the three conditions:

1. the solution is rational;
2. the solution is a convex combination of product distribution;
3. the solution is a convex combination of symmetric product distribution when the game is symmetric.

Then they constructed a symmetric game for which the only CE that satisfies the second and third conditions is irrational. Thus the algorithm must fail to find an exact CE for this game.

They observed that the reason of this failure in computing an exact CE was that the algorithm incorrectly handled certain numerical precision issues. Recall that each iteration of the ellipsoid algorithm requires an initial bounding ball with radius R and volume bound v as input such that the algorithm stops when the ellipsoid's volume is smaller than v. In the Ellipsoid Against Hope algorithm, each cut returned by the separation oracle is a convex combination of the constraints. As a result, the initial R and v may not be set appropriately, and infeasibility of (3.7) is not guaranteed.

Papadimitriou (2010) recently acknowledged the need of some minor modifications of his algorithm. He wrote as comment to a blog that "Details need to be worked out", and he was "pretty sure it should work". Possibly in consequence to this reply, at the time of this writing Stein et al. have withdrawn their paper from arXiv (its first version is still accessible).

In a paper by Jiang and Leyton-Brown (2010), recently uploaded to the arXiv, the authors show that they overcome the numerical precision issues and present a variant of the Ellipsoid Against Hope algorithm that guarantees to compute an exact CE in polynomial time. Their algorithm uses a different separation oracle that they call "the purified separation oracle". Instead of having a product distribution x to neutralize $A^\top y \leq -1$, they find a pure strategy profile s such that $(A_s)^\top y \geq 0$. Proof of existence of such s is a straightforward application of the probabilistic method. Consider $xA^\top y$ as expected value of $(A_s)^\top y$ under the distribution x. The nonnegativity of this expectation implies the existence of some s such that $(A_s)^\top y \geq 0$.

Obtaining this pure strategy s in polynomial time needs identifying the product distribution x satisfying $xA^\top y = 0$ that is used in the separation oracle of the original algorithm. For every player p, the pure strategy s_p in s is computed by iterating though all strategies in S_p to find a strategy s_p such that $[x_{(p \to s_p)} A^\top] y \geq 0$, where $x_{(p \to s_p)}$ is the product distribution in which player p plays s_p and all other players play according to x. Therefore after the algorithm stops and returns infeasibility we have a sequence of pure strategy profiles $(s^{(i)})_i$ instead of a sequence of product distributions $(x^{(i)})_i$. If we denote by $A' = AX^\top$ the matrix for which each column i corresponds to the pure strategy $s^{(i)}$ from each iteration, then we have

$$(A')^\top y \leq -1, \; y \geq 0. \tag{3.8}$$

Replace (3.6) in the original Ellipsoid Against Hope algorithm with (3.8). The precision issues are overcome because each constraint of (3.8) is also one of the constraints of (3.3), and as a result neither the maximum value of the coefficients nor the right-hand side of (3.8) is greater than that in (3.3). Therefore the initial bound and volume for (3.3) are also suitable for (3.8). This algorithm is called "the Simplified Ellipsoid Against Hope Algorithm".

The above CE computation algorithms based on the Ellipsoid algorithm with a separation oracle are possible to adapt to EFCE, due to the similarity in the existence proofs for CE and EFCE. We show this in Section 3.6.

3.4 EFCE and incentive constraints

As we pointed out in Section 3.2, there is no "sequence form" to compute an EFCE for multi-player games. The compact description holds for two-player games because the condition of perfect recall imposes strong restrictions on the player's information sets, so that the recommended move at each information set can be generated uniquely. However, with more players or chance introduced, the consistency across information sets is only a necessary condition. For this reason, our system does not use the sequence form. Instead, we compute the correlated distribution $z : S \to \mathbb{R}$ on the set of strategy profiles, and introduce the expected payoff contribution and optimal expected payoff of a move.

We make two assumptions for the EFCE:

Assumption 1: each player assumes all other players always follow their recommendations; this is a standard assumption for any equilibrium concept.

Assumption 2: When a player deviates, he gets no further information. Hence the posterior of the player at subsequent information sets is that at the last information before he deviates. This assumption can be made without loss of generality because any EFCE can be defined using reduced strategies only (von Stengel and Forges 2008, Section 2.2). Throughout this thesis, whenever we consider a strategy s_p of player p, any move of that strategy at an information set k of p that is unreachable due to an own earlier move in s_p is always assumed to be the same (say, the first) move at k which therefore assumes the role of an unspecified move "$*$" as in a reduced strategy because it carries no information.

Analogous to the incentive constraints in Theorem 3.4 for two-player games, the incentive constraints for multi-player games have three kinds of attributes: the constraints for the expected payoff contribution, the constraints for the optimal expected payoff, and the constraints for comparing the expected payoff contribution and the optimal expected payoff. We first explain the expected payoff contribution.

We introduce the auxiliary variable $u(c)$ for any $c \in C_h$ and $h \in H_p$ to denote the expected payoff contribution of c when player p gets the recommendation c at information set h, and follows all the recommendations he gets. The following definitions clarify the relationship between the relevant information sets, moves, strategies, and the leaves of the game tree.

Definition 3.5 *An information set $h \in H_p$* precedes *another information set $k \in H_q$ if and only if $p = q$ and there are nodes $u \in h$ and $v \in k$ such that u is on the path from the root to v. And $h \in H_p$* immediately precedes *$k \in H_p$ if there is a move $d \in C_h$ such that $\sigma_k = \sigma_h d$.*

We write $h < k$ (equivalently, $k > h$) if h precedes k, and $h \le k$ (equivalently, $k \ge h$) if $h = k$ or h precedes k. Note that whenever $h < k$ is written then h and k belong to the same player p. By perfect recall, $<$ is a partial order on H_p, in fact a "tree order" where the set $\{h \mid h < k\}$ for any k is linearly ordered.

When von Stengel and Forges (2008) describe the EFCE for two-player games, they use the word "precede" for any two information sets that are connected (that is, there are nodes in these information sets that are on one path from the root of the tree to a leaf), regardless of which players these information sets belong to. In this thesis, two information sets so that one "precedes" the other belong to the same player. We do this because only the relationship between information sets of the same player is important for the system that we use to describe the set of EFCE; moreover, $<$ describes an order only for information sets of one player (see von Stengel and Forges, Fig. 6, for information sets of two players with a cyclical "precedes" relationship).

Definition 3.6 *An information set $h \in H_p$ is* reachable *by a partial strategy profile $s_{-p} \in S_{-p}$ if and only if there exists s_p such that player p reaches h with positive probability at a certain stage if all players choose the moves in $s = (s_p, s_{-p})$.*

Let T be the set of leaves (or terminal nodes) of the game tree. For a node $t \in T$ and a player p, let $\sigma^p(t)$ be the sequence of moves of player p on the path from the root of the game tree to t.

Definition 3.7 *For a move c of player p, we say $t \in T$* succeeds *c if c is a move in $\sigma^p(t)$, and that t* terminates *c if c is the last move of $\sigma^p(t)$. We say that c is a* *terminal move if there exists $t \in T$ such that t terminates c. A strategy s_p of player p* agrees with *t if s_p agrees with $\sigma^p(t)$, that is, makes all moves in that sequence. A partial profile $(s_q)_{q\in Q}$ of strategies for some set of players Q* agrees with *t if each s_q agrees with t.*

We denote by $\bar{a}^p(t)$ the payoff to player p if node t is reached at the end of the game. For games with chance moves, let the expected payoff contribution be $a^p(t)$, which is $\bar{a}^p(t)$ times the product of all chance probabilities on the path from the root to t.

The following example illustrates the relationships defined above.

Example 3.8 Consider the game tree in Figure 3.1.

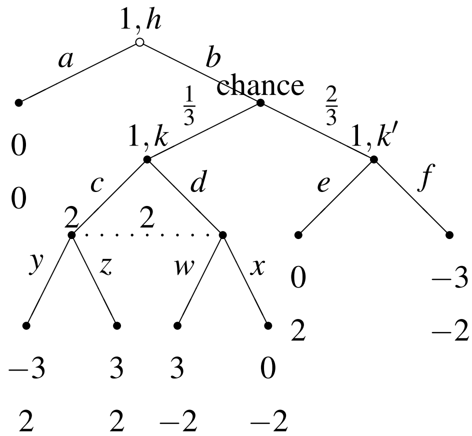


Figure 3.1: An extensive game.

The information set h *precedes* both k and k' but not the information set of player 2. Let t be the node with payoffs $(0,2)$. Then $\sigma^1(t) = be$. Also, t succeeds b and e, and terminates e. The strategies (bde) and (bce) agree with t.

For any move c of player p, the expected payoff contributions $u(c)$ are given by

$$u(c) = \sum_{t\in T\ :\ t \text{ succeeds } c} a^p(t) \sum_{s\in S\ :\ s \text{ agrees with } t} z(s). \tag{3.9}$$

where $z(s)$ is the probability distribution according to which the correlation device selects the strategy profile s.

Now we explain the incentive constraints for the optimal expected payoff.

At any information set h and any move $c \in h$, the expected payoff contribution $u(c)$ is compared with the possible payoff for the player when he deviates from his recommendation. Given a move $c \in h$ and an information set k such that $k \geq h$, we use $v(k,c)$ to denote the optimal expected payoff at k given that the player gets the recommended move c at h. It is the maximum of the payoffs for the possible moves $d \in C_k$, this optimal expected payoff is given by a payoff $a^p(t)$ where t terminates d, or given as the subsequent optimal payoffs at later information sets. This is expressed by the following inequalities:

$$v(k,c) \geq \sum_{\substack{s_p \,:\, s_p \text{ agrees} \\ \text{with } \sigma_h c}} \; \sum_{\substack{t \in T \,:\, t \\ \text{terminates } d}} \; \sum_{\substack{s_{-p} \,:\, s_{-p} \\ \text{agrees with } t}} a^p(t)\, z(s_p, s_{-p}) + \sum_{l \,:\, \sigma_l = \sigma_k d} v(l,c), \qquad \text{for all } d \in C_k. \tag{3.10}$$

Intuitively, the player follows the recommendations at all preceding information sets he arrives, and reaches the information set h when he considers deviating. From the moment that he deviates, by Assumption 2, he decides to ignore any further recommendations he would get (or, equivalently, he is not given recommendations any more). There is a path from h to anther information set k of this player. If he follows this path and gets k, the expected payoff for him is

$$\sum_{\substack{s_p \,:\, s_p \text{ agrees} \\ \text{with } \sigma_h c}} \; \sum_{\substack{t \in T \,:\, t \\ \text{terminates } d}} \; \sum_{\substack{s_{-p} \,:\, s_{-p} \\ \text{agrees with } t}} a^p(t)\, z(s_p, s_{-p}) + \sum_{l \,:\, \sigma_l = \sigma_k d} v(l,c)$$

for a certain move d he chooses at the information set k, assuming that for the subsequence information sets l, the optimal expected payoff he can get is $v(l,c)$. Thus, $v(k,c)$ is the optimal expected payoff for this player considering deviating to c at information set k.

Note also the strong similarity of (3.10) with, first, the incentive constraints in (iv) of Theorem 3.4, and secondly with the dual constraints $E^\top u \geq Ay$ of the sequence form for Nash equilibria in (2.20) where $v(k,c)$ takes the role of a dual variable u_k associated with an information set, except that these variables exist now conditional on each recommended move c. This is no coincidence, as these variables express the best possible payoff contribution at an information set k.

We complete the system characterizing the set of EFCE with the expression that compares the expected payoff contribution and the optimal expected payoff. For any player p and any move $c \in C_h$ of this player, this is given by

$$u(c) = v(h,c). \tag{3.11}$$

We prove that the incentive constraints characterize the EFCE.

Theorem 3.9 *In a perfect-recall extensive game, a probability distribution z that fulfills for all players the incentive constraints (3.9),(3.10) and (3.11) defines an EFCE. The number of these constraints that describe the set of EFCE is polynomial in the size of the game tree.*

Proof. When player p gets the recommended move c at information set h, his posterior on the behavior of the other players is given by $z(s)$ for any strategy profile $s_{-p} \in s_{-p}$. When player p deviates and considers another move d at h or at any succeeding information set k, as in (3.10), his posterior about the other players does not change because he will not get further information when he deviates. The recommendation is generated at the beginning of the game (the root of the tree), and the equilibrium conditions assumes that the other players follow the recommendations. Thus $v(h,c)$ is the optimal payoff he can get from deviating from c. □

Intuitively, at any information set h, by choosing the recommended move c, the player p gets at least as much payoff as he does from deviating to any other moves at h. Therefore, the player has no incentive to deviate.

Von Stengel and Forges (2008) mention that in general extensive form games, any EFCE is an *agent form correlated equilibrium* or AFCE, where each information set is replaced by a separate player whose actions are exactly the moves at that information set, and who receives the same payoff as the original player. However, the set of AFCE outcomes can be larger than the set of EFCE outcomes. Conceptually, this holds because the incentive constraints in an EFCE are more restrictive because the player can change his behaviour at his future information sets after a deviation; in an AFCE, that future behaviour is fixed as given by the other agents, including his own agents. Technically, this is expressed by the incentive constraints (iv) in Theorem 3.4 for two-player games, and for multi-player games by the incentive constraints (3.10).

Consider the incentive constraints (3.9), (3.10) and (3.11) that describe the set of EFCE. For every move $c \in h$, the equation (3.11) compares the expected payoff contribution of c and the optimal expected payoff at the information set h where c is given as a recommended move. The auxiliary variables $u(c)$ and $v(h,c)$ are redundant. We can remove these variables in two steps: first we substitute the redundant $v(h,c)$ (that is, the auxiliary variable $v(h,c)$ such that $c \in h$) with $u(c)$; and second we substitute $u(c)$ with the sum in (3.9). Then, the system (3.9), (3.10) and (3.11) is equivalent to a smaller system. An EFCE is a distribution z on S such that for each player p, the following constraints are satisfied.

(i) For all $h \in H_p$ and $c,d \in C_h$,

$$\sum_{t\in T\ :\ t \text{ succeeds } c} a^p(t) \sum_{s\in S\ :\ s \text{ agrees with } t} z(s) \geq \sum_{\substack{s_p\ :\ s_p \text{ agrees} \\ \text{with } \sigma_h c}} \sum_{\substack{t\in T\ :\ t \\ \text{terminates } d}} \sum_{\substack{s_{-p}\ :\ s_{-p} \\ \text{agrees with } t}} a^p(t)\, z(s_p,s_{-p}) \;+ \sum_{l\ :\ \sigma_l=\sigma_k d} v(l,c) \tag{3.12}$$

and

(ii) For all pairs (k,c) of an information set $k \in H_p$ and a move $c \in C_h$ such that $h \in H_p$ and $h < k$, and all $d \in C_k$,

$$v(k,c) \geq \sum_{\substack{s_p\ :\ s_p \text{ agrees} \\ \text{with } \sigma_h c}} \sum_{\substack{t\in T\ :\ t \\ \text{terminates } d}} \sum_{\substack{s_{-p}\ :\ s_{-p} \\ \text{agrees with } t}} a^p(t)\, z(s_p,s_{-p}) + \sum_{l\ :\ \sigma_l=\sigma_k d} v(l,c) \tag{3.13}$$

So there are a polynomial number of constraints in the systems (3.12) and (3.13). Let K be the set of all pairs (c,d) of moves $c \in C_h$ and $d \in C_k$ such that $h,k \in H_p$ and $h \leq k$. The constraints are one for a pair in K.

The smaller system of inequalities (3.12) and (3.13) also defines the set of EFCE. We use this smaller system in later sections.

3.5 The existence proof

This section presents a constructive proof of the existence of the EFCE for multi-player games with perfect recall. We do this by providing an interpretation of the dual variables in the proof, which is adapted from that for CE by Hart and Schmeidler (1989) as described above (see also Nau and McCardle 1990).

The main theorem of this section is the existence of an EFCE as stated in Theorem 3.1.

The proof of Theorem 3.1 is adapted from the proof of existence for CE by Hart and Schmeidler. For the first step, let D be the set of all pairs (k,c) of an information set $k \in H_p$ and a move $c \in C_h$ such that $h \in H_p$ and $h < k$. The variables of the combined system (3.12) and (3.13) are $z(s)$ for all $s \in S$ and $v(k,c)$ for all $(k,c) \in D$. Denote $z = (z(s))_{s \in S}$ and $v = (v(k,c))_{(k,c) \in D}$. We write the system (3.12) and (3.13) as the matrix inequality $Az + Bv \geq 0$. Consider the linear program that maximizes the sum of $z(s)$ for all $s \in S$

$$\begin{array}{llll} \text{maximize} & \sum_{s\in S} z(s) & & \\ \text{subject to} & Az & +Bv & \geq 0 \\ & z & & \geq 0 \end{array} \tag{3.14}$$

where the constraints $Az + Bv \geq 0$ are the inequalities in (3.12) and (3.13), and the variables v are the free variables that are without nonnegativity constraints. Obviously (3.14) is always feasible with the trivial solution $z = 0$ and $v = 0$. For the existence of an EFCE, it is sufficient to prove that (3.14) is unbounded. By duality, this can be achieved by showing the infeasibility of its dual system

$$\begin{array}{rcl} A^\top y & \leq & -1 \\ B^\top y & = & 0 \\ y & \geq & 0. \end{array} \tag{3.15}$$

For that purpose we need the following lemma, which we will show subsequently.

Lemma 3.10 *For every $y \geq 0$, $B^\top y = 0$, there is a product distribution x such that $x^\top A^\top y = 0$.*

Lemma 3.10 shows that a product distribution x exists to "neutralize" each purported dual solution y to (3.15).

Inspired by the CE existence proof by Hart and Schmeidler (1989), we use an auxiliary zero-sum game between two players I and II to prove Lemma 3.10. Similarly to the game for CE, the auxiliary game for EFCE is constructed such that player I chooses a strategy profile $s \in S$, and player II's payment to player I is $(A^\top y)_s$, where (which we show later) any $y \geq 0$ that fulfills $B^\top y = 0$ represents a mixed strategy of player II. A mixed strategy z of player I is then an EFCE if it guarantees nonnegative payoff to player I. For that it suffices to show that to any mixed strategy y of player II, player I has a best response that gives him nonnegative payoff, which is the product distribution x in Lemma 3.10. This replicates the argument by Hart and Schmeidler (1989).

In the auxiliary game, a *pure* strategy of player II is to choose a triple $\big(h, c, V(c)\big)$ where $c \in C_h$, and $V(c)$ is a combination of moves at $k \geq h$ (including d for $k = h$) such that $k > h$ are reachable by move d, and reachable due to subsequently chosen moves, as in a reduced strategy. Let $s^p_{V(c)}$ be the (pure) strategy of player p that chooses moves at $k \geq h$ according to $V(c)$ and at all other information sets (including those preceding h) according to s_p, and let $s' = (s^p_{V(c)}, s_{-p})$.

As in the auxiliary game for CE, it is useful to consider two payments, the first payment from player II to player I when following the strategy profile s chosen by player I, and the second payment from player I to player II according to s', that is, when s chosen by player I is followed except for the deviation of player II. If s_p does not agree with $\sigma_h c$, there is no payment. Otherwise, player II pays to player I

$$\sum_{\substack{t \in T \,:\, t \\ \text{agrees with } s}} a^p(t) \tag{3.16}$$

and player I pays to player II

$$\sum_{\substack{t \in T : t \\ \text{agrees with } s'}} a^p(t). \tag{3.17}$$

For any y such that $y \geq 0$ and $B^\top y = 0$, we want to construct a mixed strategy for player II so that player II's combined expected payment to player I, that is, (3.16) minus (3.17), is $(A^\top y)_s$ if player I chooses s. We can assume that player II plays in two steps by first choosing an information set h and a move $c \in C_h$, and then in step two chooses the combination $V(c)$.

For the given y, let

$$q_c^h = \sum_{d \in C_h} y_{c,d}^h, \tag{3.18}$$

which is the probability that player II chooses h and $c \in C_h$. (We will shortly show that it is no restriction to assume that the numbers q_c^h are indeed probabilities.) If $q_c^h = 0$, then neither c nor $V(c)$ is chosen at all. For $d \in C_h$, choose d with probability $y_{c,d}^h / q_c^h$. For information sets $l > h$ and moves $e \in C_l$, let $\sigma_l = \sigma_k d$ for the preceding information set $k \geq h$. Then if $y_{c,d}^h = 0$, then l will not be reached with $V(c)$ and no move at l has to be specified. Otherwise, choose e with probability $y_{c,e}^h / y_{c,d}^h$.

This describes the random choice of $V(c)$ by player II in the second step, after having chosen h and c, as a "partial behaviour strategy" that specifies a behaviour for all information sets k that succeed h. This requires that for any information sets h, k, l, and moves $c \in C_h$, $d \in C_k$ such that $h \leq k$ and $\sigma_k d = \sigma_l$, we must have that $y_{c,e}^h / y_{c,d}^h$ are probabilities for all $e \in C_l$, so that $\sum_{e \in C_l} y_{c,e}^h / y_{c,d}^h = 1$. Also, in the first step, q_c^h must be probabilities for all h and $c \in C_h$, so that $\sum_h \sum_{c \in C_h} q_c^h = 1$. The first condition holds because of the following lemma.

Lemma 3.11 *$B^\top y = 0$ if and only if for any h and moves $c \in C_h$, $d \in C_k$, where $k \geq h$, we have*

$$y_{c,d}^h = \sum_{e \in C_l} y_{c,e}^h, \tag{3.19}$$

where $\sigma_l = \sigma_k d$.

Proof. Notice that (3.19) is analogous to (2.1), with $y_{c,d}^h$ corresponding to $x(\sigma_k d)$, except that $y_{c,\emptyset}^h$ corresponding to $x(\emptyset)$ is not defined, and that only rows of $Ex = e$ are considered that correspond to the information set h and any succeeding information sets $k \geq h$. In particular, we do not consider the first row $x(\emptyset) = 1$ of $Ex = e$, which is the only row with a nonzero right hand side. We therefore have $E'y = 0$, with $y^h_{c,d}$ replacing $x_{\sigma_k d}$ in $Ex = e$, where E' is obtained by only keeping the rows in E that correspond to h and any succeeding information sets k. The corresponding dual constraints are $(E')^\top v' = 0$. The components of v' will be $v(k,c)$ for $k \geq h$. Comparing (3.12) and (3.13) with (2.20), which are written as $Az + Bv \geq 0$, we find that the columns of B for variables $v(k,c)$ consist of these matrices $(E')^\top$, which are constructed for each choice of h and c. The conditions (3.19) hold in each case. □

We call $B^\top y = 0$, that is, (3.19), the *consistency constraints* for the dual variables y.

For the second condition that q^h_c must be probabilities, we need

$$\sum_h \sum_{c \in C_h} q^h_c = \sum_h \sum_{c,d \in C_h} y^h_{c,d} = 1. \tag{3.20}$$

However, the dual variables $y^h_{c,d}$ in (3.15) do not necessarily sum up to 1 for $c,d \in C_h$. Similarly to the existence proof for CE, this issue can be handled by re-scaling these variables. The contradiction from a neutralizing strategy x is not affected. This holds because $A^\top y \leq -1$ is linear, so the re-scaling only means the right-hand side of these inequalities will not be -1 but a different negative number. The only case this may fail is when $\sum_h \sum_{c,d \in C_h} y^h_{c,d} = 0$. Due to the consistency constraints (3.19), this means that all $y^h_{c,d} = 0$ for all $c \in C_h$, $d \in C_k$ and $k \geq h$. However, in that case $A^\top y \leq -1$ is not fulfilled anyway.

Given that player II chooses c at h as a first step, the choice of $V(c)$ in the second step as described above defines a partial behaviour strategy by specifying a behaviour for each reachable information set $k \geq h$, which according to Kuhn's theorem is realization equivalent to any mixture of partial pure strategies $V(c)$. In fact, we have replicated the proof of Kuhn's theorem in the construction before Lemma 3.11. It implies the following observation.

Lemma 3.12 *Assume that $y \geq 0$, $B^\top y = 0$ and that y represents a probability distribution on pairs h,c with $c \in C_h$ as in (3.20). Then for $k \geq h$ and $d \in C_k$, the variable*

$y^k_{c,d}$ is the probability that player II chooses h and c and $V(c)$ so that $V(c)$ reaches k and chooses d at k.

Now suppose player I chooses s and player II employs the described strategy of choosing first h and c, and then $V(c)$. If s_p does not agree with $\sigma_h c$, there is no payment. Otherwise, the payment (3.16) from player II to player I given that player I chooses s is independent of $V(c)$ and given by

$$\sum_h \sum_{c,d \in C_h} y^h_{c,d} \sum_{\substack{t \in T \,:\, s \text{ agrees with } t, \\ t \text{ succeeds } c}} a^p(t) \,. \tag{3.21}$$

The payment from player I to player II (3.17) is determined by the terminal moves in $V(c)$ and s_{-p}. It is given by

$$\sum_{(h,c,d) \in U} y^h_{c,d} \sum_{\substack{t \in T \,:\, t \text{ terminates } d \\ \text{and } s_{-p} \text{ agrees with } t}} a^p(t) \tag{3.22}$$

where $U = \{(h,c,d) \mid c \in C_h,\ d \in C_k$ where $k \geq h$ and d is a terminal move$\}$. So the combined payoff to player I is

$$\sum_h \sum_{c,d \in C_h} y^h_{c,d} \sum_{\substack{t \in T \,:\, s \text{ agrees with } t, \\ t \text{ succeeds } c}} a^p(t) - \sum_{(h,c,d) \in U} y^h_{c,d} \sum_{\substack{t \in T \,:\, t \text{ terminates } d \\ \text{and } s_{-p} \text{ agrees with } t}} a^p(t) \,. \tag{3.23}$$

We now examine the entries of A, which are defined by (3.12) and (3.13). By (3.12),

$$\begin{aligned} u(c) &= \sum_{t \in T \,:\, t \text{ succeeds } c} a^p(t) \sum_{s \in S \,:\, s \text{ agrees with } t} z(s) \\ &= \sum_{\substack{s_p \,:\, s_p \text{ agrees} \\ \text{with } \sigma_h c}} z(s) \sum_{\substack{t \in T \,:\, s \text{ agrees with } t, \\ t \text{ succeeds } c}} a^p(t). \end{aligned}$$

The second equation holds because s_p agrees with $\sigma_h c$ if and only if there is a node t such that t succeeds c and s agrees with t. So (3.12) is equivalent to

$$\sum_{\substack{s_p \,:\, s_p \text{ agrees} \\ \text{with } \sigma_h c}} z(s)\Big[\sum_{\substack{t\in T\,:\, t \text{ terminates } d \\ \text{and } s_{-p} \text{ agrees with } t}} a^p(t) - \sum_{\substack{t\in T\,:\, s \text{ agrees with } t, \\ t \text{ succeeds } c}} a^p(t)\Big] + \sum_{l\,:\,\sigma_l=\sigma_k d} v(l,c) \geq 0. \tag{3.24}$$

Compare (3.24), which replaces (3.12), and (3.13), with (3.23). Notice that there is no payment when s_p does not agree with $\sigma_h c$, which is consistent with the observation that in (3.24) and (3.13) the entries of A corresponding to column s and row (h,c,d) are zero for s_p not agreeing with $\sigma_h c$. We find that player II's expected payment to player I (over the mixed behaviour strategy y) given that player I chooses s is indeed $(A^\top y)_s$.

It remains to show that the neutralizing mixed strategy x exists. Let us consider the product probability distribution $x = \prod_h x^h$ where x^h is the probability distribution over moves at the information set h. Since $C_k \bigcap C_h = \emptyset$ for $h \neq k$, without ambiguity, for any move $c \in C_h$ we can use $x(c)$ instead of $x^h(c)$ to denote the probability of c being chosen according to x^h.

When player II chooses $\big(h,c,V(c)\big)$ according to the mixed strategy y, it is not immediately clear how to neutralize the resulting "deviating behaviour" at an information set k which may be due to any choice of player II of an information h where $h \leq k$, and corresponding move c at h, if player I chooses a strategy profile s so that s agrees with $\sigma_h c$. The neutralizing product strategy x is stated in the following theorem.

Theorem 3.13 *Consider an information set $k \in H_p$ and assume that the product distribution x defines the behaviour of player p at every information set $h < k$. For $\sigma = \sigma_h$ or $\sigma = \sigma_k$, let $x[\sigma] = \prod_{c \text{ in } \sigma} x(c)$. Let $y \geq 0$ and $B^\top y = 0$ and (3.20) hold. For every $e,d \in C_k$ define*

$$Y_{ed} = x[\sigma_k]\, y^k_{e,d} + \sum_{h<k} x[\sigma_h] \sum_{c\in C_h} x(c)\, y^h_{c,d}\ . \tag{3.25}$$

With $n = |C_k|$, choose $x(d)$ as x_d for $d \in C_k$ according to Lemma 3.2. Suppose x is defined inductively this way for all information sets k of every player. Then $x^\top A^\top y = 0$ as claimed in Lemma 3.10.

Proof. We show that the payment in the auxiliary game is zero if player I chooses s according to the product distribution x and player II chooses $\big(h,c,V(c)\big)$ according to y.

Assume that the information sets that precede k are $h_1,\ldots,h_K$ with $h_1<\cdots<h_K=k$, and let $d\in C_k$ and $\sigma_k d=d_1\cdots d_{K-1}d_K$ where $d_j\in C_{h_j}$ (and $d_K=d$) for $1\le j\le K$. We prove that

$$x[\sigma_k d]\left(\sum_{j=1}^{K} q_{d_j}^{h_j}\right)=\sum_{j=1}^{K} x[\sigma_{h_j}]\sum_{c\in C_{h_j}} x(c)\,y_{c,d}^{h_j}\,. \tag{3.26}$$

First, suppose that $K=1$, so that there are no information sets that precede k. Then σ_k is the empty sequence and $x[\sigma_k]=1$, so that (3.26) states

$$x(d)\cdot q_d^k=\sum_{c\in C_k} x(c)\,y_{c,d}^k\ ,$$

which because $q_d^k=\sum_{e\in C_k} y_{d,e}^k$ is exactly the stationarity property (3.1) of $x(d)$ stated in Lemma 3.2. This is generalized in the following inductive proof, where we assume that (3.26) holds for $K-1$ instead of K, that is,

$$x[\sigma_k]\left(\sum_{j=1}^{K-1} q_{d_j}^{h_j}\right)=\sum_{j=1}^{K-1} x[\sigma_{h_j}]\sum_{c\in C_{h_j}} x(c)\,y_{c,d_{K-1}}^{h_j}\,. \tag{3.27}$$

We use the stationarity property of the probabilities $x(d)$ for the moves d at k according to (3.1) in Lemma 3.2, that is,

$$x(d)\sum_{e\in C_k} Y_{de}=\sum_{e\in C_k} x(e)\,Y_{ed}\,. \tag{3.28}$$

We expand the two sides of this equation using (3.25).

The left hand side of (3.28) is

$$\begin{aligned}
x(d)\sum_{e\in C_k} Y_{de} &= x(d)\sum_{e\in C_k}\left(x[\sigma_k]\,y_{d,e}^k+\sum_{h<k} x[\sigma_h]\sum_{c\in C_h} x(c)\,y_{c,e}^h\right)\\
&= x(d)x[\sigma_k]\sum_{e\in C_k} y_{d,e}^k + x(d)\sum_{j=1}^{K-1} x[\sigma_{h_j}]\sum_{c\in C_{h_j}} x(c)\sum_{e\in C_k} y_{c,e}^{h_j}\\
&= x(d)x[\sigma_k]\sum_{e\in C_k} y_{d,e}^k + x(d)\sum_{j=1}^{K-1} x[\sigma_{h_j}]\sum_{c\in C_{h_j}} x(c)y_{c,d_{K-1}}^{h_j} && \text{by (3.19)}\\
&= x[\sigma_k d]q_d^k + x(d)x[\sigma_k]\left(\sum_{j=1}^{K-1} q_{d_j}^{h_j}\right) && \text{by (3.27)}
\end{aligned}$$

which is the left hand side of (3.26).

The right hand side of (3.28) is (note that the second term in (3.25) is independent of e)

$$\begin{aligned}\sum_{e\in C_k} x(e)\,Y_{ed} &= \sum_{e\in C_k} x(e)\,x[\sigma_k]\,y^k_{e,d} + \left(\sum_{e\in C_k} x(e)\right)\sum_{j=1}^{K-1} x[\sigma_{h_j}] \sum_{c\in C_{h_j}} x(c)\,y^{h_j}_{c,d} \\ &= \sum_{j=1}^{K} x[\sigma_{h_j}] \sum_{c\in C_{h_j}} x(c)\,y^{h_j}_{c,d}\end{aligned}$$

which is the right hand side of (3.26). This completes the proof of (3.26).

Equation (3.26) is exactly the analogous property as used in the auxiliary game for a strategic form CE in the reasoning before (3.5). Namely, consider any leaf t of the game tree and the unique path from the root to t. Suppose that $\sigma_k d$ is the sequence of moves of player p on that path, where $\sigma_k d = d_1 \cdots d_K$ on the left side of (3.26) (if that sequence is empty, that is, player p does not move at all, there is no payment). According to the product strategy x of player I in the auxiliary game, that sequence is chosen with probability $x[\sigma_k d]$, and the sequence has to be chosen in order to reach the leaf p. Assume that chance and the other players choose moves in s_{-p} so that the leaf t is reached (which has a fixed probability); the probabilities considered in the following are conditioned on this event.

In the same way as argued before (3.5), the first payment $a^p(t)$ from player II to player I as one of the terms in (3.16) results if and only if player I chooses one of the moves in $\sigma_k d$ and player II chooses one of the information sets h_j and move d_j at h_j for $1 \le j \le K$. The probability for any of these disjoint events is $\sum_{j=1}^{K} q^{h_j}_{d_j}$. The combined probability of this first payment is the expression on the left hand side of (3.26).

The right hand side of (3.26) is the probability that the second payment (3.17) from player I to player II results at leaf t. This happens if player II has chosen one of the information sets h_j for $j = 1, \ldots, K$, that play has reached that information set (with probability $x[\sigma_{h_j}]$), and that player I has chosen a move c at h_j so that player II deviates from c and later chooses move d at the last information set k, according to Lemma 3.12.

By (3.26), the two probabilities are equal. This shows that x neutralizes y as claimed. □

This completes the proof of Theorem 3.1.

3.6 The algorithm for EFCE computation

In this section, we turn our existence proof in Section 3.5 into a polynomial algorithm. This algorithm is adapted from the polynomial time algorithm for CE computation, which we reviewed in Section 3.3.

In each iteration of the ellipsoid algorithm, compared with the algorithm that computes a CE, our algorithm for EFCE has an extra step to maintain the candidate solution y_i to satisfy the consistency constraints $B^\top y_i = 0$. Before the first iteration, it simplifies the system $B^\top y = 0$, $y = 0$ in the usual way by identifying basic columns of this system and expressing the corresponding basic variables as linear combinations of free variables $\bar{y}$, so that $y = \bar{B}\bar{y}$. Then the system (3.15) is equivalent to

$$\begin{array}{rcl} A^\top \bar{B}\bar{y} & \leq & -1 \\ \bar{y} & \geq & 0. \end{array} \tag{3.29}$$

We apply the ellipsoid algorithm to the system (3.29). The separation oracle becomes that for any $\bar{y} \geq 0$, there exists z such that $z[A^\top \bar{B}]\bar{y} = 0$, or for the purified separation oracle, there exists a pure strategy profile s such that $A_s\bar{B}\bar{y} \geq 0$.

Since we know that (3.29) is infeasible, the algorithm will result in recognizing the infeasibility of the system after polynomially many iterations. Thus when the algorithm halts, we have polynomially many candidate solutions y_i and for each y_i a corresponding product distribution z_i.

We now show that a convex combination, denoted $Z^\top \xi$, of these product distributions can be found in polynomial time, such that the system $AZ^\top \xi + Bv \geq 0$, $\xi \geq 0$ is unbounded. When the ellipsoid algorithm is applied to (3.29), in each iteration the inequality $(z_i^\top A^\top \bar{B})\bar{y} \leq -1$ is violated by $\bar{y}_i$. Let Z be the matrix where each row i is the product distribution z_i found by the ellipsoid algorithm. We consider the system of linear inequalities

$$[ZA^\top \bar{B}]\bar{y} \le -1, \qquad \bar{y} \ge 0. \tag{3.30}$$

Clearly, the number of variables of (3.30) is equal to that of (3.29), and is polynomial in the size of the game tree. Thus the ellipsoid algorithm is appropriate to solve (3.30) too. Apply it to (3.30). Let the initial candidate solution be $\bar{y}_0 = 0$. In each iteration i, the ith constraint of (3.30) $(z_i^\top A^\top \bar{B})\bar{y} \le -1$ is violated by the ith candidate solution y_i. Thus the algorithm will determine that (3.30) is infeasible too. That is,

$$[ZA^\top]y \le -1, \qquad y = \bar{B}\bar{y}, \qquad \bar{y} \ge 0$$

or equivalently

$$[ZA^\top]y \le -1, \qquad B^\top y = 0, \qquad y \ge 0$$

is infeasible. The dual problem

$$\text{maximize} \sum_i (\xi_A)_i \qquad \text{subject to} \quad [AZ^\top]\xi_A + B\xi_B \ge 0, \quad \xi_A \ge 0 \tag{3.31}$$

is unbounded. Here (ξ_A, ξ_B) is a partition of the variable vector ξ.

For any feasible solution ξ of (3.31), ξ_A after normalization is a probability distribution on the set of strategy profiles. The product $Z^\top \xi_A$ is a convex combination of the rows of $Z^\top$, which are the product distributions that are computed at all the iterations of the ellipsoid algorithm. Thus the nonnegative constraints $\xi_A \ge 0$ are satisfied if and only if $Z^\top \xi_A \ge 0$. Let $z = Z^\top \xi_A$, and $v = \xi_B$. The system (3.31) becomes

$$\text{maximize} \sum_s z(s) \qquad \text{subject to} \quad Az + Bv \ge 0, \quad z \ge 0$$

which is the system that characterizes an EFCE. Therefore, $(z, v) = (Z^\top \xi_A, \xi_B)$ is a nontrivial solution to (3.15) when ξ is a nontrivial solution to (3.31). Furthermore, $z = Z^\top \xi_A$ is the desired EFCE.

For the complexity of this algorithm, we examine the three conditions in the Ellipsoid Against Hope by Papadimitriou and Roughgarden (2008). We find that it satisfies that the game has polynomial number of players and polynomial expectation property for product distributions. It is not of polynomial type. But this can be

---

Algorithm: Computing One EFCE.

Initialize:	$y = \bar{B}\bar{y}$, $i = 1$,
	$\bar{y} = 0, V = n4^D I, v = (2^{D+1}\sqrt{n})^n$
Step 1:	Get x_i by Lemma 3.10 such that $[x_i^\top A^\top]\bar{B}\bar{y} = 0$.
	Compute $\tilde{V}, \tilde{v}, \tilde{\bar{y}}$
	(Or for Simplified Ellipsoid Against Hope algorithm, get the pure strategy profile s such that $A_s\bar{B}\bar{y} \geq 0$.)
Step 2:	If $\tilde{v} < 2^{-(n+1)D}$ then go to Step 3.
	Otherwise replace $V, v, \bar{y}$ with $\tilde{V}, \tilde{v}, \tilde{\bar{y}}$,
	$i = i + 1$ and return to Step 1.
Step 3:	Let $X = [x_i]_i$, solve the problem (3.31) for ξ.
	For a non-trivial, normalized (by dividing by $\sum \xi_A$) ξ_A, let $z = X^\top \xi_A$.
Output:	z, represented in polynomial size as $z = X^\top \xi_A$.

---

overcome. This holds because given a fixed behaviour of player II, the mixed strategy x in Step 1 can be computed in polynomial time. Therefore this is a polynomial time algorithm for EFCE computation, and we have Theorem 1.14.

Appendix: The ellipsoid algorithm

Here we explain the ellipsoid method we use in Algorithm 3.6 to compute one EFCE. Our description is based on Papadimitriou and Steiglitz (1998). Although this polynomial algorithm works for *linear strict inequalities* (LSI) only, one can prove that the complexity of linear programming (LP), linear inequalities (LI) and LSI are equivalent.

We first briefly review the definition of LP, LI, and LSI, and the relationship between the complexity of these problems.

Definition 3.14 Linear programming *(LP) is the following computational problem: Given an integer $m \times n$ matrix A, an m-vector b and and n-vector c, either* *(a)* *Find a rational n-vector x such that* $x \geq 0$, $Ax = b$,*and* $c^\top x$ *is minimized subject to these conditions, or*

(b) *Report that there is no n-vector x such that* $x \geq 0$ *and* $Ax = b$*, or*

(c) *Report that the set* $\{c^\top x \mid Ax = b, x \geq 0\}$ *has no lower bound.*

The problem of linear inequalities *(LI) is defined as follows:*

> *Given an integer* $m \times n$ *matrix A and m-vector b, is there an n-vector x such that* $Ax \leq 0$*?*

Linear strict inequalities *(LSI) is the following problem:*

> *Given an* $m \times n$ *integer matrix A and m-vector b, is there an n-vector x such that* $Ax < b$*?*

Consider an $m \times n$ LP

$$\begin{array}{lrcl} \text{maximize} & c^\top x & & \\ \text{subject to} & Ax & = & b \\ & x & \geq & 0 \end{array} \tag{A.32}$$

The following lemma states that the rational numbers in a basic feasible solution to such an LP have an upper bound.

Lemma 3.15 (Papadimitriou and Steiglitz 1998) *The basic feasible solutions of (A.32) are n-vectors of rational numbers, both the absolute value and the denominators of which are bounded by* 2^L*, where*

$$L = mn + \lceil log|P| \rceil,$$

and P is the product of the nonzero coefficients appearing in A, b, and c.

It is easy to induce an upper bound for a LI from that of LP. Consider the LI $Ax \leq b$. It is equivalent to the following LP

$$\begin{array}{lrcl} \text{maximize} & 0 & & \\ \text{subject to} & Ax + Iy & = & b \\ & y & \geq & 0 \end{array}$$

The vector in the objective function is $c = 0$. In the constraints, I is the $m \times m$ identity matrix and for the component y_i of the m-vector y is the vector of slack variables.

Theorem 3.16 *The solutions of the LI $Ax \leq b$ are n-vectors of rational numbers, both the absolute value and denominators are bounded by 2^L, where*

$$L = mn + \lceil log|P| \rceil,$$

and P is the product of the nonzero coefficients appearing in A and b.

For a LSI $Ax < b$, consider the LI

$$Ax \leq b - \varepsilon \tag{A.33}$$

where ε is an m-vector and each component $\varepsilon_i = 2^{-2L}$. Obviously any solution of the of the LSI $Ax < b$ is also a solution of (A.33). Therefore, the upper bound of (A.33) is the upper bound of the LI too.

The following result describes the relationship between the complexity of these problems.

Theorem 3.17 *The following statements are equivalent:*

(i) There is a polynomial algorithm for LP.

(ii) There is a polynomial algorithm for LI.

(iii) There is a polynomial algorithm for LSI.

The ellipsoid algorithm works for LSI system. It determines whether the system has a solution in polynomial time. The idea is to start with an initial ellipsoid that contains the solutions of the given system, if there is any solution. Since the upper bound is given in Lemma 3.15, as long as the volume of the initial ellipsoid is large enough, one solution can be contained. The algorithm then proceeds in iterations. In each iteration, it first checks whether the center point of the ellipsoid is a solution of the LSI system. If yes, then the algorithm stops with output this solution. Otherwise the ellipsoid shrinks by a certain ratio, yet the solution is maintained in it, and the next iteration starts. After polynomial iterations, either a solution is found, or the ellipsoid becomes too small to contain a solution, when the algorithm reports that the system is infeasible. The algorithm is shown below:

The Ellipsoid Algorithm.

Input: An $m \times n$ system of linear strict inequalities $Ax < b$, of size L.

Initialize: Set $j = 0$, $t_0 = 0$, $B_0 = n^2 2^{2L} \cdot I$.

Step 1: If t_j is a solution to $Ax < b$ then return t_j;
If $j > K = 16n(n+1)L$ then return "no".

Step 2: Choose any inequality $a^\top t_j \geq b$ in $Ax < b$ that is violated by t_j. Set

$$\begin{aligned} t_{j+1} &= t_j - \frac{1}{n+1} \frac{B_j a}{\sqrt{a^\top B_j a}} \\ B_{j+1} &= \frac{n^2}{n^2-1}[B_j - \frac{2}{n+1}\frac{(B_j a)(B_j a)^\top}{a^\top B_j a}] \\ j &= j+1 \end{aligned}$$

and go to Step 1.

Output: An n-vector x such that $Ax < b$, if such a vector exists; "no" otherwise.

In the initial step, the initial ellipsoid is

$$\begin{aligned} T(S_n) &= \{y \in \mathbb{R}^n \mid (y - t_0)^\top B_0^{-1}(y - t_0) \leq 1\} \\ &= \{y \in \mathbb{R}^n \mid y^\top B_0^{-1} y \leq 1\} \\ &= \{y \in \mathbb{R}^n \mid n^{-2} 2^{-2L} y^\top y \leq 1\} \end{aligned}$$

is the round sphere of radius $n \cdot 2^L$, hence contains the set of solutions of the problem, if there is any. The initial candidate of solution is the origin which is the center of the round sphere. In each iteration, the algorithm checks in Step 1 if the candidate of the solution, which is the center of the current ellipsoid, is a solution of the system. Then a sphere $a^\top t_j \geq b$ is chosen in Step 2. Since $a^\top t_j \geq b$ is a constraint of the system, there must be a solution in the intersection of this sphere and the current ellipsoid. In this step, the ellipsoid shrinks with the intersection maintained in the new ellipsoid. When the new iteration starts, the counter j is checked to determine if the current ellipsoid is large enough to contain the solution.

If the system is infeasible, then the ellipsoid will become too small to contain a solution before polynomial steps. This is because of the following two theorems, one for the ratio by which the ellipsoid shrinks at each iteration, and the other for the minimum volume of the intersection of the set of solutions of LSI and the sphere of the initial round of the algorithm.

Theorem 3.18 (Papadimitriou and Steiglitz 1998) *Let B_j be a positive definite matrix, let $t_j \in \mathbb{R}^n$, and let a be any nonzero n-vector. Let B_{j+1} and t_{j+1} be as in Step 2 of the ellipsoid algorithm. Then the following hold.*

(a) *B_{j+1} is positive-definite (or, equivalently, $E_{j+1} = \{x \in \mathbb{R}^n \mid (x - t_{j+1})^\top B_{j+1}^{-1}(x - t_{j+1}) \leq 1\}$ is an ellipsoid.)*

(b) *The semiellipsoid*

$$\frac{1}{2}E_j[a] = \{x \in \mathbb{R}^n \mid (x - t_j)^\top B_j^{-1}(x - t_j) \leq 1, a^\top(x - t_j) \leq 0\}$$

is a subset of E_{j+1}.

(c) *The volumes of E_j and E_{j+1} satisfy*

$$\frac{vol(E_{j+1})}{vol(E_j)} < 2^{-1/2(n+1)}$$

Theorem 3.19 *If an LSI system of size L has a solution, then the set of solutions within the sphere $\|x\| \leq n2^L$ has volume at least $2^{-(n+2)L}$.*

References

Audet, C., P. Hansen, B. Jaumard, and G. Savard (2001), Enumeration of all extreme equilibria of bimatrix games. SIAM Journal on Scientific Computing 23, 323–338.

Aumann, R. J. (1974), Subjectivity and correlation in randomized strategies. Journal of Mathematical Economics 1, 67–96.

Avis, D. (2000), lrs: A revised implementation of the reverse search vertex enumeration algorithm. In: Polytopes – Combinatorics and Computation, eds. G. Kalai and G. Ziegler, Birkhäuser-Verlag, DMV Seminar Band 29, 177–198.

Avis, D. (2011), User's guide for lrs. `http://cgm.cs.mcgill.ca/~avis/C/lrs.html`. Accessed 14 January 2011.

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

Avis, D., R. Savani, G. Rosenberg, and B. von Stengel (2010), Enumeration of Nash equilibria for two-player games. Economic Theory 42, 9–37.

Chen, X., and X. Deng (2006), Settling the complexity of 2-player Nash-equilibrium. In: FOCS, 261–272, 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)

Chu, F., and J. Halpern (2001), On the NP-completeness of finding an optimal strategy in games with common payoffs. International Journal of Game Theory 30, 99–106.

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

Daskalakis, C., P. Goldberg, and C. Papadimitriou (2006), The complexity of computing a Nash equilibrium. in: 38th Annual ACM Symposium on Theory of Computing, STOC, 71–78

Dhillon, A., and J. F. Mertens (1996), Perfect correlated equilibria. Journal of Economic Theory 68, 279–302.

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

Hart, S., and D. Schmeidler (1989), Existence of correlated equilibria. Mathematics of Operations Research 14, 18–25.

Huang, W., and B. von Stengel (2008), Computing an extensive-form correlated equilibrium in polynomial time. In: Workshop on Internet and Network Economics (WINE 2008), eds. C. Papadimitriou and S. Zhang, Lecture Notes in Computer Science 5385, Springer Verlag, Berlin, pp. 506–513

Jiang, A., and K. Leyton-Brown (2010), Polynomial-time computation of exact correlated equilibrium in compact games. `http://arxiv.org/abs/1011.0253`. Accessed 17 January 2011.

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

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

Kuhn, H. W. (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.

Millham, C. B. (1974), On Nash subsets of bimatrix games. Naval Research Logistics Quarterly 21, 307–317.

Myerson, R. B. (1986), Multistage games with communication. Econometrica 54, 323–358.

Myerson, R. B. (1991), Game Theory – Analysis of Conflict. Harvard Univ. Press, Cambridge, Mass.

Myerson, R. B. (1997), Dual reduction and elementary games. Games and Economic Behavior 21, 183–202.

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

Nau, R. F., and K. F. McCardle (1990), Coherent behavior in noncooperative games. Journal of Economic Theory 50, 424–444.

Papadimitriou, C. H. (2010), Comments on the blog post “Computation of correlated equilibrium for concise games”. Accessed 27 October 2010. `http://agtb.wordpress.com/2010/10/15/computation-of-correlated-equilibrium-for-concise-games/`.

Papadimitriou, C. H., and T. Roughgarden (2008), Computing correlated equilibria in multi-player games. J. ACM 55, Article 14.

Papadimitriou, C. H., and K. Steiglitz (1998), Combinatorial Optimization: Algorithms and Complexity. Prentice-Hall, New York.

Rosenberg, G. (2005), Enumeration of all extreme equilibria of bimatrix games with integer pivoting and improved degeneracy check. Research Report LSE-CDAM-2005-18, London School of Economics.

Selten, R. (1975), Reexamination of the perfectness concept for equilibrium points in extensive games. International Journal of Game Theory 4, 25–55.

Stein, N. D., P. A. Parrilo, and A. Ozdaglar (2010), Exchangeable equilibria contradict exactness of the Papadimitriou-Roughgarden algorithm. LIDS Technical Report 2852, MIT. `http://arxiv.org/abs/1010.2871v1`. Accessed 17 January 2011.

von Stengel, B. (1996), Efficient computation of behavior strategies. Games and Economic Behavior 14, 247–259.

von Stengel, B. (2001), Computational complexity of correlated equilibria for extensive games. Research Report LSE-CDAM-2001-03, London School of Economics.

von Stengel, B. (2002), Computing equilibria for two-person games. Chapter 45, Handbook of Game Theory, Vol. 3, eds. R. J. Aumann and S. Hart, North-Holland, Amsterdam, 1723–1759.

von Stengel, B., and F. Forges (2008), Extensive form correlated equilibrium: Definition and computational complexity. Mathematics of Operations Research 33, 1002–1022.

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

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