International Game Theory Review, Vol. 11, No. 4 (2009) 437–451

DOI: 10.1142/S021919890900242X

A NEW SEQUENCE FORM APPROACH FOR THE ENUMERATION AND REFINEMENT OF ALL EXTREME NASH EQUILIBRIA FOR EXTENSIVE FORM GAMES

CHARLES AUDET

GERAD and Département de mathématiques et de génie industriel
École Polytechnique de Montréal
C.P. 6079, Succ. Centre-ville
Montréal (Québec) Canada H3C 3A7
charles.audet@gerad.ca

SLIM BELHAIZA

Département de mathmatiques et génie industriel
École Polytechnique de Montréal
C.P. 6079, Succ. Centre-ville
Montréal (Québec) Canada H3C 3A7
slim.belhaiza@gerad.ca

PIERRE HANSEN

GERAD and Méthodes quantitatives de gestion
HEC Montréal, 3000
Chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7
pierre.hansen@gerad.ca

This paper presents two new results on the enumeration of all extreme equilibria of the sequence form of a two person extensive game. The sequence form of an extensive game is expressed, for the first time to our knowledge, as a parametric linear $0-1$ program. Considering $Ext(P)$ as the set of all of the sequence form extreme Nash equilibria and $Ext(Q)$ as the set of all the parametric linear $0-1$ program extreme points, we show that $Ext(P) \subseteq Ext(Q)$. Using exact arithmetics classes, the algorithm EχMIP Belhaiza (2002); Audet *et al.* (2006) is extended to enumerate all elements of $Ext(Q)$. A small procedure is then applied in order to obtain all elements of $Ext(P)$.

Keywords: Sequence form; extensive game; Nash equilibrium; extreme equilibrium; enumeration; EχMIP algorithm.

1. Introduction

Games have been described and represented in different forms. The *extensive* form and the *strategic* (normal) form are the most important forms of games. One goal of game theorists is to try to predict the way a game will be played by computing an equilibrium or by enumerating all of its extreme equilibria. In this scope, previous work (Belhaiza (2002), Audet *et al.* (2006)) studied the question of complete enumeration of extreme equilibria for strategic form games, i.e. bimatrix and polymatrix games. The extensive form is known to be the most richly structured way to describe game situations. The standard definition of an extensive game is due to Kuhn (Kuhn, 1953). An extensive form game is usually represented by a finite tree where the players moves are represented by branches (edges).

Computing equilibrium for an extensive game has generally been achieved through its conversion to strategic form. Each combination of moves of a player in the extensive form is then represented by a strategy. To avoid exponential increase of the game's description and to reduce the complexity introduced by this conversion, Wilson (1972) and Koller and Megiddo (1996) propose computations that use mixed strategies with small support. Romanovski (1962), Selten (1988), Koller and Megiddo (1992) and von Stengel (1996) use a sequence form approach in which pure strategies are replaced by move sequences.

This work introduces a new mixed $0-1$ parametrized formulation of an extensive game, and extends the EχMIP algorithm for strategic form games (Audet *et al.* (2006)) to extensive form games. This formulation is based on the sequence form of an extensive game. The paper is divided as follows. Section 2 presents extensive form games in both their strategic form and sequence form representations. Section 3 details the new mixed $0 - 1$ parametrized formulation. The new implementation of the EχMIP algorithm is described in Sec. 4 and illustrated on some test problems. For one of the examples, the sequence form approach allows the identification of a subgame perfect equilibrium.

2. Extensive Form Games

In an extensive game, game states are represented using tree nodes. The root node indicates the beginning of the game and each leaf (terminal node) indicates the end of the game. At the end of the game, each player receives a payoff. In such a game tree, the nonterminal nodes are considered as the decision nodes and the player's moves are attributed to the outgoing branches (edges) of the decision node. As some events are determined by chance, a node where the next branch is determined by a random mechanism is called a chance node. Hence, each possible sequence of moves is represented by a path from the root to one of the terminal nodes. When the game is played, the path that represents the sequence of moves of the players is commonly called *path of play*.

Kuhn (1953) introduced a partition of the decision nodes into information sets. Considering all nodes in an information set as nodes belonging to the same player and having the same moves, a deciding player knows only the information set but not the particular node he is at.

In a n-person extensive game, each nonterminal node has a player label in the set $\{0, 1, 2, \ldots, n\}$ and nodes with a 0 label are called chance nodes. Thus, the nodes with the player-label i define the set of nodes controlled by player i. Hence, and following Kuhn's (Kuhn, 1953) partition, each node controlled by a player has a second label indicating the information set that the player would consider if the path of play reaches this node. Furthermore, each possible move at a node controlled by a player has a move label. Considering any pair of nodes k and l that have the same player and information set labels, for each alternative at the node k there must be exactly one alternative representing the same move at node l.

It is also generally assumed that all players have *perfect recall* in an extensive game. This last condition means that each deciding player remembers all the information he knew earlier in the game, including his past moves. Myerson (1997) gives a more detailed expression to this perfect recall condition.

In a game tree, a *subgame* is defined as a subtree including all information sets that contain a node of the subtree. Extensive game equilibria can recursively be found by considering subgames. However, in this scope a distinction has to be made between games with *perfect information* and games with *imperfect information*. In a perfect information game, each information set is a singleton and each node is a root of a subgame, i.e. no two nodes have the same information set. Backward induction can then be used to compute an equilibrium for a game with perfect information.

2.1. *Strategic form representation*

For each player i, the set of information sets is denoted H_i. The information sets are denoted by h and the set of moves at h by C_h. Nash equilibria of an extensive game are generally defined as equilibria of its *strategic form*, in particular for games without subgames.

In an extensive game, a pure strategy of player i is defined as a deterministic move at each information set. Thus, a pure strategy is an element of $\Pi_{h \in H_i} C_h$. Moves at information sets that cannot be reached due to an earlier own move are identified in the *reduced strategic form*.

A player may also have parallel information sets that can not be distinguished by his own earlier moves. This happens for example when a player receives information about an earlier move by another player. The reduced strategic form is generally exponential in the game tree size and this has limited the use of extensive games.

By applying the Lemke-Howson algorithm to the strategic form of an extensive game, Wilson (1972) shows that the number of pure strategies can be large during the computation. Koller and Megiddo (1996) show that by introducing a system of linear equations for realization weights of the game tree leaves and by using a *basis crashing* subroutine, a small support of the computed mixed strategies can be maintained.

Moreover, while the best response subroutine in Wilson's algorithm (Wilson, 1972) requires perfect recall, Koller and Megiddo (1996) propose a method to enumerate small supports in a way that can be extended to extensive games without perfect recall.

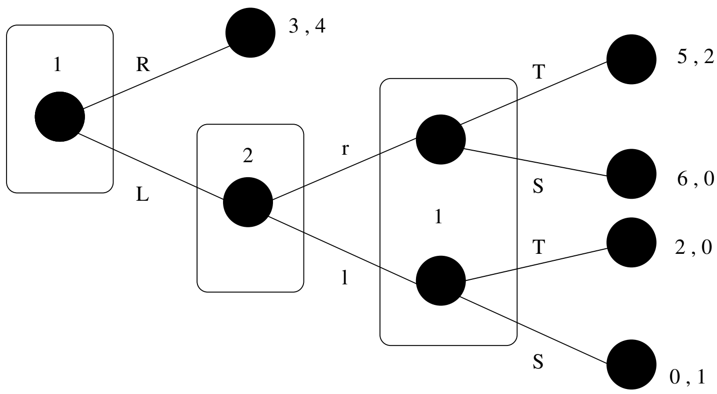

Fig. 1. Two person extensive game. (von Stengel *et al.* (2002))

Figure 1 illustrates a two person extensive game taken from von Stengel *et al.* (2002). Player 1 plays once or twice while player 2 plays at most once. For his first move, player 1 has two alternatives L or R. If player 1 chooses to play L at his first move, player 2 will have two alternatives l or r. If player 2 decides to play l or r, player 1 will have two other alternatives S or T. If player 1 decides to play R at his first play, the game ends and player 1 receives 3 as final payoff while player 2 receives 4 as final payoff. In Fig. 1, the pure strategies of player 1 are the move combinations $\langle L, S\rangle$, $\langle L, T\rangle$, $\langle R, S\rangle$ and $\langle R, T\rangle$. It is possible to replace $\langle R, S\rangle$ and $\langle R, T\rangle$ by a single reduced strategy that represents the top row of the subsequent strategic form. Hence, the reduced strategic form of this game is

$$A_r = \begin{pmatrix} 3 & 3 \\ 0 & 6 \\ 2 & 5 \end{pmatrix} \quad B_r = \begin{pmatrix} 4 & 4 \\ 1 & 0 \\ 0 & 2 \end{pmatrix}.$$

By applying the EχMIP algorithm (Belhaiza (2002), Audet *et al.* (2006)) to this reduced strategic form, three extreme equilibria are enumerated. x_1 and x_2 represent the vectors of mixed strategies of players 1 and 2, while α and β correspond to the payoffs of the players.

This game has two extreme equilibria $Eq1$ and $Eq2$ which are not subgame perfect. These restrictions to the subgame starting at player 2's node define no equilibria to this subgame.

Table 1. Reduced strategic form Extreme Nash Equilibria.

Eq.		x_1		x_2		α_1	α_2
1	1	0	0	2/3	1/3	3	4
2	1	0	0	1	0	3	4
3	0	2/3	1/3	1/3	2/3	4	2/3

Elimination of all redundant strategies leads to a *full reduced strategic representation* of the extensive game. Furthermore, elimination of all strongly dominated strategies yields a residual strategic game easier to solve in order to compute all extreme equilibria.

2.2. *Sequence form representation*

The use of *sequences* of moves instead of pure strategies is also possible in order to compute equilibria for extensive form games. As the extensive game is represented by a tree, there exists a unique path linking the root node to any node of such a tree. This path defines a sequence of moves for player i. Assuming that each player i has perfect recall means that each pair of nodes in an information set h in H_i correspond to the same sequence for player i.

In most papers, the sequence of moves leading to h is denoted σ_h and the set of move sequences is denoted S_i for each player i. Any sequence of moves $\sigma \in S_i$ can either be equal to the empty sequence $\emptyset$ or only given by its last move at the information set $h \in H_i$, which means that $\sigma = \sigma_h c$. This leads to the following definition

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

Following this definition, each player will have a number of sequences that does not exceed the number of nodes in the tree. As detailed by von Stengel (1996, 2002), the *sequence form* of an extensive game is similar to its strategic form reduction. The only difference between these two conversions is that the sequence form uses sequences instead of pure strategies which leads to a more compact description of the original game.

3. Mixed $0-1$ Formulation

For a given player i, a behavior strategy β is obtained by probabilities $\beta(c)$ for his moves $c \in C_h$ such that $\beta(c) \geq 0$ and $\sum_{c \in C_h} \beta(c) = 1$ for each $h \in H_i$. Behavior strategy's definition can be extended to the sequences $\sigma \in S_i$ simply by the following formulation

$$\beta\left[c\right] = \prod_{c \in \sigma} \beta(c).$$

In this context, a pure strategy π for a given player is a kind of behavior strategy with $\pi(c) \in \{0, 1\}$ for all moves c, which means that $\pi\left[\sigma\right] \in \{0, 1\}$ for each $\sigma \in S_i$. Thus, a mixed strategy μ corresponds to a probability $\mu(\pi)$ to every pure strategy π of a player i. The *realization probabilities* of playing the sequences $\sigma \in S_i$ are defined as follows

$$\mu\left[\sigma\right] = \sum_{\forall \pi} \mu(\pi)\pi\left[\sigma\right].$$

For a player i, a realization plan of μ is then denoted $x(\sigma) = \mu\left[\sigma\right]$ for $\sigma \in S_i$. For a given player i, x_i is the realization plan of a mixed strategy if and only if the

FA 1

April 8, 2010 15:51 WSPC/0219-1989 151-IGTR 00242

following conditions are satisfied

$$\begin{aligned} x(\emptyset) &= 1, \\ \sum_{c \in C_h} x(\sigma_h c) &= x(\sigma_h), h \in H_i, \\ x(\sigma) &\geq 0, \quad \forall\, \sigma \in S_i. \end{aligned}$$

Koller and Megiddo (1992) refer to these conditions using realization probabilities of the game tree. Denoting $x_i = (x_\sigma)_{\sigma \in S_i}$, these conditions can be reformulated for each player i as follows

$$x_i \geq 0, \tag{1}$$

$$E_i x_i = e_i, \tag{2}$$

where E_i is a well chosen matrix and $e_i = (1, 0 \ldots, 0)^t$, both with $1 + |H_i|$ rows.

In the example illustrated by Fig. 1, the two following payoff matrices A and B give the different payoffs for each player considering his set of sequences:

$$A = \begin{bmatrix} & & \\ & & \\ 3 & & \\ & 0 & 6 \\ & 2 & 5 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} & & \\ & & \\ 4 & & \\ & 1 & 0 \\ & 0 & 2 \end{bmatrix}.$$

Hence, the sets of sequences are respectively $S_1 = \{\emptyset, L, R, LS, LT\}$ and $S_2 = \{\emptyset, l, r\}$. Following conditions (1) and (2), E_1, E_2, e_1 and e_2 are written as follows

$$E_1 = \begin{bmatrix} 1 & & & & \\ -1 & 1 & 1 & & \\ & -1 & & 1 & 1 \end{bmatrix}, \quad e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad E_2 = \begin{bmatrix} 1 & & \\ -1 & 1 & 1 \end{bmatrix}, \quad e_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Koller and Megiddo (1992) show that two mixed strategies μ and μ' of player i are realization equivalent if and only if they have the same realization plan, i.e. $\mu[\sigma] = \mu'[\sigma]$ for all $\sigma \in S_i$.

Moreover, Kuhn (1953) show that for a player with perfect recall, any mixed strategy is realization equivalent to a behavioral strategy.

Furthermore, Romanovskii (1962) and von Stengel (1996) demonstrate that the equilibria of a two person game in extensive form with perfect recall are simultaneous solutions of the following pair of parametrized linear programs

$$\begin{aligned} &\max_{x_1} x_1^t A x_2 && \text{and} && \max_{x_2} x_1^t B x_2 \\ &\text{s.t. } E_1 x_1 = e_1, && && \text{s.t } E_2 x_2 = e_2, \\ &\quad x_1 \geq 0, && && \quad x_2 \geq 0. \end{aligned} \tag{3}$$

Where E_1 and E_2 are matrices with all elements equal to 1, 0 or -1. Each matrix E_1 or E_2, have as many columns as the number of sequences of play and as many lines as the number of linked sequences sets, for the corresponding player.

e_1 and e_2 are single columns with the same number of lines as E_1 and E_2, with the first element equal to 1 and all other elements equal to 0.

The dual formulations of linear programs (3) are expressed as follows

$$\begin{array}{ll} \min\limits_{\alpha_1} e_1^t\alpha_1 \quad \text{and} & \min\limits_{\alpha_2} e_2^t\alpha_2 \\ \text{s.t. } E_1^t\alpha \geq Ax_2, & \text{s.t. } \alpha_2^t E_2 \geq x_1^t B. \end{array} \tag{4}$$

The complementarity constraints obtained from (3) and (4) are

$$x_1^t(E_1^t\alpha_1 - Ax_2) = 0 \quad \text{and} \quad (\alpha_2^t E_2 - x_1^t B)x_2 = 0. \tag{5}$$

Using these complementarity conditions, von Stengel *et al.* (2002) define an algorithm able to compute normal form perfect equilibria for two-person games.

By introducing two binary vectors u_1 and u_2 as detailed in Audet *et al.* (2006), with u_1 and u_2 having as many lines as the number of sequences of the corresponding player, the complementarity conditions can be linearized as follows, with (3) and (4) satisfied

$$x_1^t(E_1^t\alpha - Ax_2) = 0 \Leftrightarrow \begin{array}{l} x_1 + u_1 \quad\quad \leq \mathbb{1}, \\ E_1^t\alpha - Ax_2 \leq L_1u_1, \end{array} \tag{6}$$

and

$$(\alpha_2^t E_2 - x^t B)x_2 = 0 \Leftrightarrow \begin{array}{l} x_2 + u_2 \quad\quad \leq \mathbb{1}, \\ \alpha_2^t E_2 - x_1^t B \leq L_2u_2. \end{array} \tag{7}$$

where $\mathbb{1}$ is a vector with all elements equal to one, and L_1 and L_2 are small scalars equal to the difference between the largest and the least element of each payoff matrix, as described in Audet *et al.* (2006). On one hand, observe that for any player i, if its i^th binary variable is equal to 1 then its i^th continuous variable is equal to 0 and the complementary slackness condition associated is satisfied. On the other hand, if its i^th binary variable is equal to 0 then the complementary slackness condition associated is also satisfied.

We propose to achieve the enumeration of all extreme equilibria of a two-persons extensive game by enumerating all extreme points of a set Q defined by the following conditions

$$\begin{array}{ll} E_1x_1 = e_1, & E_2x_2 = e_2, \\ E_1^t\alpha_1 \geq Ax_2, & \alpha_2^t E_2 \geq x_1^t B, \\ x_1 + u_1 \leq \mathbb{1}, & x_2 + u_2 \leq \mathbb{1}, \\ E_1^t\alpha - Ax_2 \leq L_1u_1, & \alpha_2^t E_2 - x_1^t B \leq L_2u_2, \\ x_1 \geq 0, & x_2 \geq 0, \\ u_1 \in \{0,1\} & u_2 \in \{0,1\}. \end{array} \tag{8}$$

Let n_1 and n_2 be the number of sequences of players 1 and 2, respectively. It follows that each binary combination of u_1 and u_2 defines a polytope and that Q is the set of all these disjoint polytopes.

Theorem 1. *Given a two-person extensive game, let the set of all of its equilibria be* $P = \{X = (x_1, x_2) : (x_1, x_2) \text{ is an equilibrium}\}$ *and* $Ext(P)$ *the set of its extreme equilibria.*

Let Q *be the set of the solutions to conditions* (8)*and* $Ext(Q)$ *the set of the extreme points of the polytopes of* Q.

Then,

$$P = Proj_x(Q) \quad and \quad Ext(P) \subseteq Proj_x(Ext(Q)).$$

Hence, the projection of Q *on the* x*-space yields* P*, any element of* P *defines at least an element of* Q *and any element of* $Ext(P)$ *can be obtained by a projection of an element of* $Ext(Q)$.

Proof. Romanovski (1962) and von Stengel (1996) show that any element $X = (x_1, x_2)$ of P is such that x_1 and x_2 are simultaneous optimal solutions of the pair of problems (3). We then defined the dual variables α and β in the linear programs (4) and introduced the binary variables u_1 and u_2 in order to satisfy the complementary slackness conditions (6) and (7). It means that any element of P is the projection over the x-space of at least one element of Q, thus $P \subseteq Proj_x(Q)$. It also implies that the projection of any element of Q over the x-space is an equilibrium of this sequence form, which means that $Proj_x(Q) \subseteq P$. Then, $Proj_x(Q) = P$. Furthermore, $Ext(P) \subseteq P$ implies that $Ext(P) \subseteq Proj_x(Q)$.

Moreover, any element of $Ext(P)$ is an equilibrium and can not be expressed as a convex combination of other equilibria. Suppose that an element $X \in Ext(P)$ is the projection of $q = (X, \alpha, u) \in Q$, such that $q \notin Ext(Q)$. Then, q can be expressed as a convex combination of at least two elements of $Ext(Q)$: $q = \sum_i \lambda^i q^i$, such that $\forall i : q^i = (X^i, \alpha^i, u^i)$, $q^i \in Ext(Q)$, $\lambda^i > 0$ and $\sum \lambda^i = 1$.

Then, $q = (\sum_i \lambda^i X^i, \sum_i \lambda^i \alpha^i, \sum_i \lambda^i u^i) = (X, \alpha, u)$. Which means that $X = \sum_i \lambda^i X^i$, with $\forall i$: $X^i = (x_1^i, x_2^i) \in P$. Knowing that $X \in Ext(P)$, we obtain $X^i = X$, $\forall i$. Hence, we may conclude that any $X \in Ext(P)$ is also the projection of at least one element of $Ext(Q)$. Thus $Ext(P) \subseteq Proj_x(Ext(Q))$. □

At this point, one can wonder if the projection on the x-space of every extreme point of Q is an extreme equilibrium of the sequence form of a 2-person extensive game. Two small examples in Sec. 4 illustrate that the answer to this question is NO: it is possible that the inclusion $Ext(P) \subset Proj_x(Ext(Q))$ is strict.

The EχMIP algorithm (Belhaiza (2002), Audet *et al.* (2006)) can be applied to the mixed $0-1$ program subject to the conditions (8) associated to a sequence form of a 2-person extensive game, in order to compute all its extreme points. The mixed $0-1$ program could have any linear objective function of the variables α_1, α_2, x_1, x_2, $u_{,1}$ and u_2. Typically we use the objective function $\alpha_1^1 + \alpha_2^1$.

By applying this new formulation to the game illustrated in Fig. 1, the following mixed $0-1$ program is obtained:

$$
\begin{array}{lll}
\max\limits_{X,\alpha,u} & \alpha_1^1+\alpha_2^1 & \\
& & \\
\text{s.t.} & x_1^1=1, & x_2^1=1, \\
& -x_1^1+x_1^2+x_1^3=0, & -x_2^1+x_2^2+x_2^3=0, \\
& -x_1^2+x_1^4+x_1^5=0, & \\
& -\alpha_1^1+\alpha_1^2\le 0, & -\alpha_2^1+\alpha_2^2+4x_1^3\le 0, \\
& -\alpha_1^2+\alpha_1^3\le 0, & -\alpha_2^2+x_1^4\le 0, \\
& -\alpha_1^2+3x_2^1\le 0, & -\alpha_2^2+2x_1^5\le 0, \\
& -\alpha_1^3+6x_2^3\le 0, & \\
& -\alpha_1^3+2x_2^2+5x_2^3\le 0, & \\
& x_1^1+u_1^1\le 1, & x_2^1+u_2^1\le 1, \\
& x_1^2+u_1^2\le 1, & x_2^2+u_2^2\le 1, \\
& x_1^3+u_1^3\le 1, & x_2^3+u_2^3\le 1, \\
& x_1^4+u_1^4\le 1, & \\
& x_1^5+u_1^5\le 1, & \\
& \alpha_1^1-\alpha_1^2\le 6u_1^1, & \alpha_2^1-\alpha_1^2-4x_1^3\le 4u_2^1, \\
& \alpha_1^2-\alpha_1^3\le 6u_1^2, & \alpha_2^2-x_1^4\le 4u_2^2, \\
& \alpha_1^2-3x_2^1\le 6u_1^3, & \alpha_2^2-2x_1^5\le 4u_2^3, \\
& \alpha_1^3-6x_2^3\le 6u_1^4, & \\
& \alpha_1^3-2x_2^2-5x_2^3\le 6u_1^5, & \\
& x_1\ge 0, & x_2\ge 0, \\
& u_1 \text{ and } u_2 & \text{binary vectors.}
\end{array}
$$

4. EχMIP in Exact Arithmetics

A new version of the EχMIP algorithm Belhaiza (2002), Audet *et al.* (2006) was implemented in C++. While the former version used Cplex, this new version uses new exact arithmetic classes. The algorithm explores a binary search tree obtained by forcing in each step one of the inequality constraints in a primal or dual LP to be tight, as required by the complementary slackness condition.

4.1. *EχMIP implementation in exact arithmetics*

In our implementation of exact arithmetic, data is always stored using rationals. A rational is a pair of integers, a numerator and a denominator. The exact arithmetic classes consist of, a *BigInteger* class, a *Rational* class, a *Simplex* class and a *Node* class.

The BigInteger class defines the new type of integers to be used during the enumeration of the extreme equilibria. This class also overloads the elementary operators for these big integers. During the implementation of EχMIP we observed that it may happen, that the numerator or denominator of a rational exceeds the value of the largest representable integer INTMAX. There is no doubt that the use of this class increases the overall enumeration time. However, the use of exact arithmetics represented a very interesting challenge which made computing time not be considered as the main objective of this work.

The Rational class is based on the Biginteger class. A rational consists of two big integers, a numerator and a denominator. After overloading the elementary operations for these rationals, a Greatest Common denominator function is applied.

The Simplex class defines a *Dictionary* (Chvátal, 1998) and the set of simplex algorithm (Dantzig, 1951) operations that will be applied to this dictionary in order to find an extreme equilibrium. A dictionary contains an array of rationals. These rationals represent the coefficients of the variables in the dictionary.

The Node class defines a framework for the algorithm EχMIP. Each node contains the current dictionary and a pointer to its father. This class contains branching methods that permits to obtain a certain number of sons nodes from a father node.

In general, the EχMIP algorithm can now be used in order to enumerate all extreme equilibria of a bimatrix game, a three person polymatrix game, or the sequence form of a two person extensive game. The *XGame Solver* software uses this implementation of the EχMIP algorithm and is available on its site http://www.XGame-Solver.net for free download.

4.2. *EχMIP algorithm*

The EχMIP algorithm explores a tree. This tree is constructed from a root node R, containing the initial mixed $0-1$ program, by *principal branching* on the binary variables u and by *secondary branching* on the continuous variables x. A current node C contains a problem to be solved using the *Simplex* algorithm (Dantzig, 1951).

Once a solution is found, the extreme point is stored and a secondary branching is made to check if the binary combination of variables u could give other extreme points. Then, from the father node on the principal tree, a principal branching is added in order to explore other combinations of binary variables. The algorithm can then be formally stated.

Step *a*. Initialization.
Let

— P; Initial mixed $0-1$ linear problem.
— X; Set of P's continuous variables.
— U; Set of P's binary variables.
— $E=\emptyset$; Set of extreme equilibria.

— $N = 0$; Depth level in the principal tree.
— R; Principal tree root node.
— C; Current node.
— x_i^k; Continuous variable associated to player i, $(i = 1, 2)$, k^{th} sequence.
— u_i^k; Binary variable associated to player i $(i = 1, 2)$, k^{th} sequence.

Take $C = R$ and *go to step b*.

Step *b*. Solving and Memorizing.
If $N \leq |X|$, solve current node problem.
If the problem is infeasible, *Go to step d*.
Else, let $\hat{e}$ be the solution obtained; If $\hat{e} \notin E$, add $\hat{e}$ to E. *Go to step c*.

Step *c*. Secondary Branching.
If the current node C belongs to the principal tree:

— Fix binary variables vector $\hat{u}$ $(\hat{u} \in U)$ at its value in $\hat{e}$.
— For all $x_i^k \in X$, such that $\hat{x}_i^k > 0$, Add the branch $x_i^k = 0$ and *Go to step b*.

Step *d*. Principal Branching.
If the current node belongs to the principal tree, no extreme point could be found from this node or its sons.
Else, return to the father node in the principal tree and choose a binary variable $u_i^k \in U$, on which there is no branching in the preceding nodes:
Let $p = N + 1$,
If $p \leq |X|$ set $N = p$:

— Add the branch $u_i^k = 0$, *Go to step b*.
— Add the branch $u_i^k = 1$, *Go to step b*.

Else, *Go to step e*.

Step *e*. End.
The set $|E|$ contains all extreme points of the mixed $0 - 1$ program.

In our paper on the Enumeration of all Extreme Equilibria for Bimatrix and Polymatix games, Audet *et al.* (2006), we prove that the EχMIP algorithm enumerates all the extreme points of the associated mixed $0-1$ programs. In fact, by capitalizing on the Simplex algorithm (Dantzig, 1951) propriety of always returning an extreme point, if there is an optimal and bounded solution, EχMIP permits to compute all the extreme points satisfying the complementary slackness conditions (5).

By principal branching, $E\chi$-MIP explores all binary variables combinations involved in one or more extreme points and by branching on binary variables till maximum depth equals the overall number of strategies involved in the game.

By secondary branching, $E\chi$MIP enumerates all extreme points that could be obtained from a binary variables combination $\hat{u}$ by fixing the combination of binary variables $\hat{u}$ and by adding branches $x_i^k = 0$. Therefore, this branching enumerates from $\hat{u}$ all extreme equilibria where some complementary slackness conditions are satisfied from both sides, x_i^k and $u_i^k = 0$.

The algorithm explores all possible ways to satisfy complementary conditions and if $\hat{e}$ is an extreme point, there exists necessarily a path in the tree generated by $E\chi$-MIP leading to $\hat{e}$.

4.3. *EχMIP on some examples*

By applying the EχMIP algorithm to von Stengel's (von Stengel *et al.* (2002)) game illustrated in Fig. 1, three extreme equilibria are found 2. One can notice that equilibria 1 and 3 have the same vectors X but have different values of α_1 and α_2.

The EχMIP algorithm is also applied to the example in Fig. 2.

The reduced strategic form of this game yields the pure strategies $\langle T\rangle$, $\langle M\rangle$, $\langle BB\rangle$, $\langle BT\rangle$ for player 1 and $\langle R\rangle$, $\langle LR\rangle$, $\langle LL\rangle$ for player 2. The reduced payoff

Table 2. Extreme points of Q: von Stengel (von Stengel *et al.* (2002)).

Eq.	X			x_1			x_2				α_1		α_2	
1		1	0	1	0	0	1	1	0	3	3	2	4	0
2		1	1	0	2/3	1/3	1	1/3	2/3	4	4	4	2/3	2/3
3	$= X^1$	1	0	1	0	0	1	1	0	3	3	3	4	0
4		1	0	1	0	0	1	2/3	1/3	3	3	3	4	0

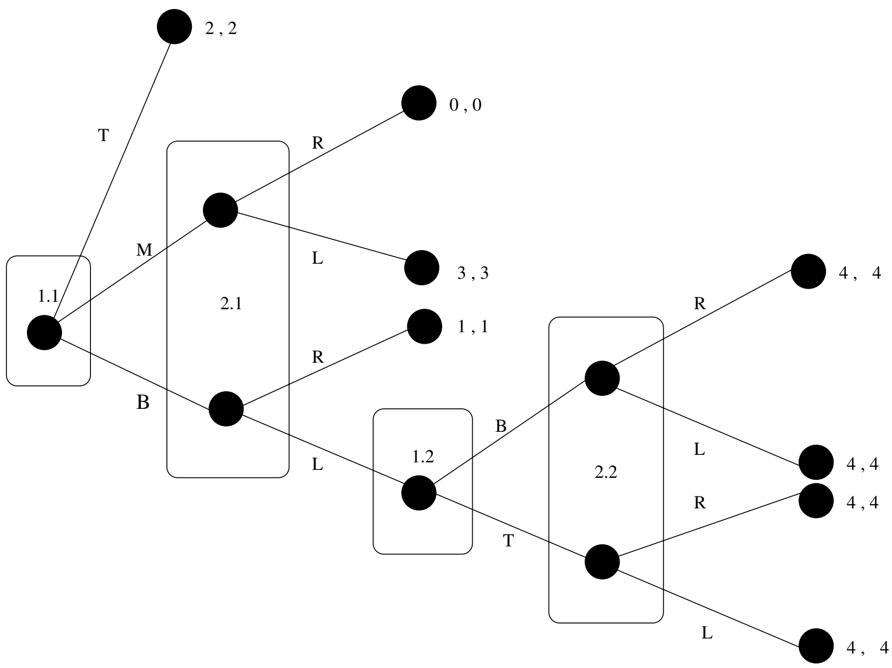

Fig. 2. Kohlberg and Mertens.

matrices are

$$A_r = \begin{pmatrix} 2 & 2 & 2 \\ 0 & 3 & 3 \\ 1 & -4 & 4 \\ 1 & 4 & -4 \end{pmatrix} \quad B_r = \begin{pmatrix} 2 & 2 & 2 \\ 0 & 3 & 3 \\ 1 & 4 & -4 \\ 1 & -4 & 4 \end{pmatrix}.$$

By applying the $E\chi MIP$ algorithm (Belhaiza (2002), Audet *et al.* (2006)) to this reduced strategic form, seven extreme Nash equilibria are enumerated.

While considering the sequence form of this game, the sequences of player 1 are $S_1 = \{\emptyset, T, M, B, BB, BT\}$ and the sequences of player 2 are $S_2 = (\emptyset, R, L, LR, LL)$. Ten extreme points are enumerated for the $0-1$ mixed integer program associated to the sequence form of this game.

Equilibria 4 and 5 correspond to the same extreme equilibrium but have different vectors of dual variables. Moreover, the vector of mixed strategies X^2 of equilibrium 2 could be obtained by the convex combination $(\frac{1}{2}, \frac{1}{2})$ of those of equilibria 1 and 3.

Equilibrium 2 is enumerated as an extreme point of the mixed integer formulation because it has a different vector of dual variables α_1. However $Proj_{(x)}(q^2) \notin Ext(P)$.

The additional information, present in Table 4 but not in Table 3 can be used for further equilibria refinement. Equilibrium 2 is found to be *subgame perfect*. Player 2 plays the mixed strategy $1/2 - 1/2$ between L and R, in the "matching pennies"

Table 3. Reduced strategic form extreme Nash equilibria (Kohlberg and Mertens (1986)).

Eq.			x_1			x_2		α_1	α_2
1	0	1	0	0	0	7/8	1/8	3	3
2	0	1	0	0	0	1/8	7/8	3	3
3	1	0	0	0	1	0	0	2	2
4	1	0	0	0	2/3	0	1/3	2	2
5	1	0	0	0	2/3	1/3	0	2	2
6	1	0	0	0	1/3	13/24	1/8	2	2
7	1	0	0	0	1/3	1/8	13/24	2	2

Table 4. Extreme points of Q: Kohlberg and Mertens (Kohlberg and Mertens (1986)).

Eq.	X			x_1						x_2			α_1			α_2	
1		1	0	1	0	0	0	1	0	1	7/8	1/8	3	3	3	3	3	0
2	$= \frac{1}{2}(X^1 + X^3)$	1	0	1	0	0	0	1	0	1	1/2	1/2	3	3	0	3	3	0
3		1	0	1	0	0	0	1	0	1	1/8	7/8	3	3	3	3	3	0
4		1	1	0	0	0	0	1	1	0	0	0	2	2	0	2	0	0
5	$= X^4$	1	1	0	0	0	0	1	1	0	0	0	2	2	1	2	0	0
6		1	1	0	0	0	0	1	2/3	1/3	1/3	0	2	2	4/3	2	0	0
7		1	1	0	0	0	0	1	2/3	1/3	0	1/3	2	2	4/3	2	0	0
8		1	1	0	0	0	0	1	1/3	2/3	13/24	1/8	2	2	5/3	2	0	0
9	$= \frac{1}{2}(X^8 + X^{10})$	1	1	0	0	0	0	1	1/3	2/3	1/3	1/3	2	2	0	2	0	0
10		1	1	0	0	0	0	1	1/3	2/3	1/8	13/24	2	2	5/3	2	0	0

subgame. This choice correponds to the unique equilibrium of this subgame. The reason is that in this equilibrium, the dual variable for the second information set 1.2 of player 1 has value 0 (which is the payoff in the subgame) because *both* moves T and B of player 1 at this information set, corresponding to the sequences BT and BB, have tight constraints relative to the dual variable. This is due to one of the binary choices of the algorithm. This subgame perfect equilibrium was not identified while solving the strategic form.

Furthermore, the mixed strategies $7/8 - 1/8$ for L and R in the equilibria q^1 and q^3 give player 1 an expected payoff of 3 at his second information set 1.2. This produces a tight constraint for only one of the sequences BT or BB, and the payoff 3 also makes player 1 indifferent between B (which is not chosen) and the equilibrium choice M at this first information set 1.1. This is the reason why this defines an extreme equilibrium. Similar observations hold for equilibrium q^9 when compared to equilibria q^8 and q^{10}.

This example shows that when using the sequence form of the extensive game, extreme subgame perfect equilibria can be found while they might be ignored when only strategic form is used. The sequence form provides more information about the extensive game than the strategic form.

5. Discussion

This work contains a survey of the main results on equilibria computation for games in extensive form. The strategic form and the sequence form conversions were described and a mixed $0-1$ formulation of the sequence form of a two person extensive game was presented for the first time. This work has shown that the algorithm EχMIP Audet *et al.* (2006) can easily be applied to the strategic form and to the sequence form representations of a two person extensive game in order to enumerate all its extreme Nash equilibria. Furthermore, the solution under the sequence form may generate some subgame perfect equilibria that would be missed under the strategic form.

References

Audet, A., Belhaiza, S. and Hansen, P. [2006] Enumeration of all extreme equilibria in game theory: Bimatrix and polymatrix games, *Journal of Optimization Theory and Applications* **129**(3) (June, 2006) 349–372.

Audet, C., Hansen, P., Jaumard, B. and Savard, G. [1998] Complete enumeration of equilibria for two-person games in strategic and sequence forms, *Eight International Symposium on Dynamic Games and Applications*, Maastricht, Netherlands.

Belhaiza, S. [2002] Énumération des équilibres extrêmes en théorie des jeux : jeux bimatriciels et polymatriciels, *M.Sc. Thesis, HEC Montréal.*

Chvátal, V. [1998] *Linear Programming* (New York, Freeman).

Dantzig, G. B. [1951] Maximization of a linear function of variables subject to linear inequalities, *Chap. XXI of "Activity Analysis of Production and Allocation"*, (ed.) T. C. Koopmans (John Wiley, New York), pp. 359–373.

Kohlberg, E. and Mertens, J. F. [1986] On the strategic stability of Equilibria, *Econometrica* **54**, 1003–1038.

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

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

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. Tucker, Annals of Mathematics Studies **28** (Princeton Univ. Press, Princeton, 1953), pp. 193–216.

Myerson, R. B. [1997] *Game Theory: Analysis of Conflict* (Harvard University Press, Cambridge-Massachusetts, London-England).

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

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

von Stengel, B. [1996] Efficient computation of behavior strategies, *Games and Economic Behavior* **14**, 220–246.

von Stengel, B., Van Den Elzen, A. and Talman, D. [2002] Computing normal form perfect equilibria for extensive two-person games, *Econometrica*, **70**, 693–715.

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